

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE
APPLIQUÉES

PAR
JULIEN DE CARUFEL

POLYFORMES ET POLYCUBES ARBRE PLEINEMENT FEUILLUS

NOVEMBRE 2018

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Alexandre Blondin Massé, directeur de mémoire

Professeur au département d'informatique de l'Université du Québec à Montréal

M. Sébastien Tremblay

Professeur au département de mathématiques et d'informatique de l'Université du Québec
à Trois-Rivières

M. Gilbert Labelle

Professeur émérite au département de mathématique de l'Université du Québec À Montréal

Remerciements

Je tiens à remercier Alain Goupil et Alexandre Blondin Massé pour toutes les opportunités de recherche et de voyage.

Merci à mes parents pour le support.

Merci à ceux qui méritent de se faire remercier, pour les raisons pour lesquelles ils le méritent.

Résumé

Dans ce mémoire, on étudie les polyformes et les polycubes arbre pleinement feuillus. Ces objets sont obtenus en maximisant le nombre de feuilles dans un polyforme ou un polycube arbre de taille fixé. Le problème des polyformes et polyominoes arbre pleinement feuillus est de trouver le nombre maximal de feuilles dans un polyforme ou un polycube arbre. L'étude de ces objets du point de vue de la théorie des graphes permet de trouver une solution au problème et de le généraliser à des graphes quelconques.

En ce sens, on trouve d'abord les fonctions feuille des polyformes et des polycubes. Cette fonction dans les entiers naturels donne le nombre maximal de feuilles que peut avoir un objet de taille donnée. On utilise ensuite les polyformes et polycubes arbre enracinés figés pour montrer l'exactitude des expressions des fonctions feuille et pour caractériser les arbres pleinement feuillus.

Finalement, on s'intéresse aux arbres saturés, qui, en plus d'avoir un nombre maximal de feuilles, ont aussi une taille donnant un ratio du nombre de feuilles sur le nombre de cellules maximal. On trouve des fonctions permettant de deviner la structure géométrique des arbres saturés. Dans le cas des polyhexes et des polytriangles, on énumère les arbres saturés.

Abstract

In this master's thesis, we study fully leafed tree-like polyforms and polycubes. These objects are found by maximising the number of leaves in a tree-like polyform or polycube of fixed size. The Fully Leafed Induced asks whether there exists an induced subtree of size n with k leaves in a given graph G . The use of graph theoretic definitions allows to find fully leafed tree-like polyforms and polycubes and to generalize them to any graph.

We first find the leaf functions for polyforms and polycubes. These functions take a positive integer which is the size of a tree-like polyform or polycube and returns the maximal number of leaves that a tree-like polyform or polycube can have. We then use grounded root tree-like polyforms and polycubes to prove that the conjectured leaf functions are exact and to find the structure of fully leafed tree-like polyforms and polycubes.

Finally, we study saturated tree-like polyforms and polycubes, which have a maximal number of leaves but also maximises the ratio of the number of leaves on the total number of cells. We describe functions that allow us to guess the geometric structure of tree-like polyforms and polycubes. In the case of polyhexes and polyiamonds, we are also able to completely enumerate all saturated trees.

Table des matières

Présentation du jury	ii
Remerciements	iii
Résumé	iv
Abstract	v
Table des matières	vi
Liste des tableaux	viii
Table des figures	ix
Introduction	2
1 Préliminaires	4
1.1 Théorie des graphes	5
1.1.1 Définitions préliminaires	5
1.1.2 Automorphismes de graphe	9
1.1.3 Graphes infinis	11
1.2 Polyominos	15
1.2.1 Graphe d'adjacence	17
1.2.2 Dénombrement	21
1.3 Polyhexes et polytriangles	23
1.3.1 Graphe d'adjacence	24

1.3.2	Dénombrement	27
1.4	Polycubes	28
1.4.1	Graphe d'adjacence	29
1.4.2	Dénombrement	32
2	Sous-arbres induits pleinement feuillus	33
2.1	Fonction feuille	34
2.2	Méthode de preuve	40
2.3	Sous-arbres induits enracinés figés	43
2.4	Saturation	58
3	Polyominos pleinement feuillus	65
3.1	Fonction feuille	67
3.2	Polyomino arbre enraciné figé	74
3.3	Saturation	83
4	Polyhexes et polytriangles	93
4.1	Fonctions feuille	94
4.2	Polyhexes et polytriangles enracinés figés	100
4.3	Saturation	106
4.4	Énumération	110
5	Polycubes pleinement feuillus	116
5.1	Fonction feuille	118
5.2	Polycubes arbre enracinés figés	122
5.3	Polycubes saturés	135
	Conclusion	140
	Bibliographie	144

Liste des tableaux

1.1	Nombre de polyominos fixés et libres	22
1.2	Nombre de polyhexes et polytriangles fixés et libres	28
1.3	Nombre de polycubes fixés et libres	32
2.1	Valeurs de la fonction feuille du graphe G de la figure 2.1	36
2.2	Valeurs de la fonction feuille du graphe W_7 de la figure 2.2	37
3.1	Valeurs de ℓ_{\square} pour n entre 2 et 19	69
3.2	Valeurs de $\Delta_{\ell_{\square}}(k)$ pour k entre 1 et 6	73
3.3	Valeurs de $\sigma_{L_{\square}}(m)$ pour m de 1 à 6	85
3.4	Nombre de polyominos arbre pleinement feuillus à translation près	92
4.1	Premières valeurs de $\ell_{\Delta}(n)$ et $\ell_{\square}(n)$	97
4.2	Premières valeurs de la fonction d'écart des polytriangles et polyhexes	100
4.3	Valeurs de $\sigma_{L_{\Delta}}(m)$ pour m de 1 à 6	107
5.1	Valeurs de $\ell_{\boxplus}(n)$ pour n de 2 à 19	121
5.2	Premières valeurs de $\Delta_{\ell_{\boxplus}}$	128
5.3	Valeurs de la fonction d'écart saturé dans les polycubes	136
5.4	Nombre de 4-arbres	140

Table des figures

1.1	Représentation d'un graphe simple G dans le plan	6
1.2	Code de couleur pour les degrés des sommets dans un graphe	7
1.3	Exemple de cycle dans un graphe et d'un graphe qui est un arbre	8
1.4	Un sous-graphe induit et un sous-arbre induit dans un graphe G	9
1.5	Exemple d'un isomorphisme entre deux graphes	10
1.6	Exemple d'un automorphisme de graphe	10
1.7	Une partie de la grille carrée	12
1.8	Une partie de la grille triangulaire	13
1.9	Une partie de la grille hexagonale	14
1.10	Une partie de la grille cubique	15
1.11	Polyominos libres et fixés d'aire 1, 2 et 3	17
1.12	Exemple de cycle et de degré dans un polyomino	18
1.13	La correspondance entre les polyominos et les sous-graphes induits de G_{\square} . . .	20
1.14	Choix d'un représentant dans une classe de polyominos à translation près . . .	21
1.15	Exemples de polyhexe	24
1.16	Exemples de polytriangle	24
1.17	Exemple de polytriangles fixés et libres	26
1.18	Correspondance entre polytriangles et sous-graphes induits de la grille hexagonale	27
1.19	Polycubes de taille 7 et 10	28
1.20	Polycubes fixés équivalents à symétrie près	29
1.21	Correspondance entre un sommet de la grille cubique et une cellule dans un polycube	31

2.1	Sous-arbres induits pleinement feuillus d'un graphe	36
2.2	Sous-arbres induits pleinement feuillus du graphe W_7	37
2.3	Illustration des démonstrations de la proposition 2.1.1	39
2.4	Espace des fonctions feuille potentielles	40
2.5	Exemples de sous-arbre induit enraciné figé	45
2.6	Exemple de greffe de sous-arbres induits enracinés figés	46
2.7	Grefe du sous-arbre induit enraciné figé identité	46
2.8	Exemple de la factorisation d'un arbre en une paire branche, co-branche	49
2.9	Substitution d'une branche dans un sous-arbre induit	50
2.10	Les trois enfants directs d'un sous-arbre induit enraciné figé	52
2.11	La coque d'un sous-graphe induit de la grille carrée	54
2.12	Exemple de l'arborescence des degrés d'un sous-arbre induit enraciné figé . . .	57
2.13	Deux polyominos arbre enracinés figés distincts ayant la même notation	57
3.1	Paires de cellules adjacentes pouvant exister dans un polyomino arbre	66
3.2	Les onze polyominos arbre d'aire 5 à symétrie près	67
3.3	Les sept polyominos arbre d'aire 2, 3 et 4	68
3.4	Polyominos arbre pleinement feuillus d'aire 6, 7, 8 et 9	68
3.5	Graphe de ℓ_{\square}	70
3.6	Famille maximale conjecturée	71
3.7	Construction récursive d'un polyomino pleinement feuillus d'aire 11	72
3.8	Graphe de $\ell_{\square}(n)$ illustrant les premières valeurs de $\Delta_{\ell_{\square}}(k)$	73
3.9	Un polyomino arbre enraciné figé	74
3.10	Les quatre polyominos arbre enracinés figés identité à translation près	75
3.11	Grefe bien définie entre deux polyominos arbre enracinés figés	76
3.12	Grefe non définie entre deux polyominos arbre enracinés figés	76
3.13	Grefe du polyomino arbre enraciné figé identité	76
3.14	Exemple de branche dans un polyomino arbre	77
3.15	Substitution dans un polyomino	78
3.16	Les polyominos arbre enracinés figés potentiellement abondants atomiques . . .	79

3.17	Polyominos arbre enracinés figés potentiellement abondants de profondeur 2 . . .	80
3.18	Polyominos arbre dont les branches sont des polyominos arbre enracinés figés abondants dont la profondeur est au plus 1 à symétrie près	82
3.19	Graphe des bornes de la fonction L_{\square}	84
3.20	Déconstruction récursive d'un polyomino arbre saturé	87
3.21	Application de l'opérateur croix ϕ sur un polyomino	88
3.22	Image de deux cellules adjacentes sous l'opérateur croix ϕ	89
3.23	Opérateur croix vu comme la greffe de croix	90
3.24	Application répétée de l'opérateur croix	91
4.1	Degré maximal dans un polyhexe arbre	94
4.2	Polyhexes arbre de taille 2 à 4	95
4.3	Polytriangles arbre de taille 2 à 4	95
4.4	Polyhexes arbre pleinement feuillus d'aire 5 à 8	95
4.5	Polytriangles arbre pleinement feuillus d'aire 5 à 8	96
4.6	Graphique de ℓ_{Δ} et ℓ_{\square}	97
4.7	Famille de polyhexes et polytriangles arbre pleinement feuillus décrite au lemme 4.1.1	99
4.8	Exemples de polyhexes et polytriangles arbre enracinés figés	100
4.9	Exemple d'une greffe entre deux polyhexes arbre enracinés figés	101
4.10	Les 12 arbre enracinés figés identités	101
4.11	Exemple d'une paire branche, co-branche dans un polyhexe arbre	102
4.12	Exemple de substitution dans un polyhexe	102
4.13	Polyhexes et polytriangles arbre enracinés figés de profondeur 1 et 2	104
4.14	Polyhexes dont les branches sont abondantes et de profondeur inférieure à 2 . .	105
4.15	Graphe des fonctions bornant L_{Δ}	107
4.16	Cas de la preuve de la proposition 4.4.1	111
4.17	Polytriangles saturés	111
4.18	Polyhexes saturés	112
4.19	Symétries des polyhexes saturés	113

4.20	Fonction ϕ entre polytriangles et polyhexes saturés	114
4.21	Exemple de la non surjectivité de la fonction ϕ sur les polytriangles	115
5.1	Différentes façons de dessiner un polycube	117
5.2	Paires de cellules de degrés maximaux	118
5.3	Polycubes arbre de taille 2 et 3	119
5.4	Polycubes arbre pleinement feuillus de taille 4, 5, 6 et 7	119
5.5	L'hypothèse initiale de fonction feuille et un contre-exemple ayant plus de feuilles	120
5.6	La super-antenne de taille 41 ayant 28 feuilles et ses cellules intérieures	120
5.7	Graphe de la fonction $\ell_{\square}(n)$	122
5.8	Un polycube arbre enraciné figé	122
5.9	Exemple de greffe de polycubes arbre enracinés figés	123
5.10	Factorisation d'un polycube arbre en une paire branche, co-branche	123
5.11	Exemple de substitution de polycubes arbre enracinés figés	124
5.12	Méthode de construction de la famille maximale des polycubes	126
5.13	Polycube pleinement feuillu de taille 100	127
5.14	Arborescence des polycubes arbre enracinés figés abondants dans les polycubes selon la profondeur	134
5.15	L'unique polycube arbre enraciné figé récursivement saturé dans les polycubes et sa substitution	137
5.16	Les 4-croix planaires et un 4-arbre	138
5.17	Les polycubes utilisés dans la fonction ϕ	139
5.18	Bijection entre les 4-arbres et les polycubes saturés	139
5.19	Vaisseau spatial dont la structure est un polycube arbre saturé [Hé18]	143

Introduction

Dans l'ensemble de ce mémoire, les polyformes et les polycubes sont étudiés. Un polyforme est défini à partir d'un pavage régulier, qui est une partition du plan où toutes les parties sont des copies du même polygone régulier. Par exemple, il est possible de paver le plan avec des carrés unitaires, on appelle ce pavage le pavage carré. En choisissant un sous-ensemble connexe du pavage carré, on obtient un polyomino, qui est un type de polyforme.

Il existe trois pavages réguliers du plan, il existe donc trois familles de polyformes. On peut paver le plan avec des triangles équilatéraux, des carrés ou des hexagones réguliers. Les polyformes associés à ces pavages sont respectivement appelés des polytriangles, des polyominos et des polyhexes.

Les polyformes peuvent être généralisés de différentes façons. Dans des dimensions supérieures, il existe une généralisation des polyformes en dimension trois. L'unique pavage régulier de l'espace tridimensionnel est celui formé de cubes. Une figure formée d'un sous-ensemble connexe de ce pavage est appelée un polycube.

Ces quatre familles de structures sont celles qui sont étudiées dans ce mémoire, mais les polyformes peuvent être généralisés de plusieurs autres façons. Il est par exemple possible d'étudier des pavages dans des dimensions quelconques. Il existe un analogue du carré et du cube dans toutes les dimensions supérieures à trois qui est appelé le d -hypercube et qui pave l'espace de dimension d . Un sous-ensemble connexe d'un pavage formé d'hypercubes de dimension d est appelé un d -polyhypercube.

Sans considération pour les dimensions, il est possible de former des structures analogues à des polyformes dans des graphes quelconques. Cette dernière généralisation permet d'obtenir toutes les familles décrites plus haut.

Au chapitre 1, on présente des concepts de base de la théorie des graphes et on définit les polyominos, polytriangles, polyhexes et polycubes. On utilise une définition des polyformes et des polycubes basée sur la théorie des graphes qui nous permet de généraliser le problème et les définitions.

Au chapitre 2, les objets relatifs à l'étude des arbres pleinement feuillus sont définis et le problème des arbres pleinement feuillus est présenté dans le cas général. L'utilisation des fonctions feuille est centrale à ce chapitre. On étudie les sous-arbres induits pleinement feuillus enracinés figés afin de montrer l'exactitude des fonctions feuille que l'on conjecture et afin de caractériser les arbres pleinement feuillus.

Au chapitre 3, on étudie les polyominos arbre pleinement feuillus. On trouve la fonction feuille de la grille carrée, et on montre son exactitude. On caractérise ensuite les polyominos arbre saturés en trouvant une fonction entre les polyominos arbres saturés et des polyominos arbres plus petits. Cette fonction permet de connaître le nombre exact de polyominos arbres saturés, dans la mesure où l'on connaît le nombre de polyominos arbres.

Au chapitre 4, on étudie les polyhexes et polytriangles arbre pleinement feuillus. Comme au chapitre précédent, on trouve et on montre l'exactitude de la fonction feuille de la grille hexagonale et de la grille triangulaire. On montre que ces deux fonctions feuille sont égales. Dans le cas de ces grilles, les arbres saturés ont une structure très restreinte qui permet leur énumération. Une fonction bijective est décrite entre les polyhexes saturés et les polytriangles saturés.

Dans le chapitre 5, on étudie les polycubes arbre pleinement feuillus. On trouve la fonction feuille, qui est plus complexe que dans les cas précédents. Encore une fois, on caractérise les polycubes arbre saturés. On trouve aussi une fonction bijective entre les polycubes arbre saturés et un sous-ensemble des polycubes appelés des 4-arbres.

Chapitre 1

Préliminaires

C'est une histoire vraie, plus ou moins.

— Kurt Vonnegut Jr

Dans ce chapitre, on définit certains objets qui sont utilisés durant tout le mémoire. Les définitions utilisées ne sont pas spécifiques à l'étude des polyformes et des polycubes pleinement feuillus.

En particulier, à la section 1.1 les graphes sont introduits. Dans l'ensemble de ce mémoire, les objets sont étudiés sous l'axe de la théorie des graphes.

Dans les sections 1.2, 1.3 et 1.4, on définit les polyominos, les polytriangles, les polyhexes et les polycubes. On utilise d'abord une définition géométrique pour introduire les objets intuitivement, puis on utilise la théorie des graphes pour obtenir une définition plus générale.

1.1 Théorie des graphes

Plusieurs concepts de la théorie des graphes sont utiles à l'étude des polyformes et des polycubes. L'utilisation des définitions et des résultats de ce domaine permet de simplifier les raisonnements. La plupart des définitions sont tirées du livre *Théorie des graphes* de Jacques Labelle [Lab81].

1.1.1 Définitions préliminaires

L'étude des graphes est centrale à plusieurs domaines des mathématiques, et plus précisément à la combinatoire. Un graphe caractérise en général les relations entre des paires d'objets, ce qui fait en sorte que les graphes apparaissent fréquemment à des endroits inattendus.

Définition 1.1.1 – Graphe [Lab81]

Un graphe simple G est formé de deux ensembles : un ensemble X , appelé l'ensemble des sommets de G et un ensemble A de paires de sommets, appelé l'ensemble des arêtes de G . On a donc $A \subset \mathcal{P}_2(X)$. On écrit souvent $G = (X, A) = (X(G), A(G))$.

L'ensemble $\mathcal{P}_2(X)$ est l'ensemble de tous les sous-ensembles à deux éléments de X .

Lorsqu'aucune précision n'est apportée, le terme **graphe** est utilisé pour signifier graphe simple. On illustre généralement un graphe par un dessin dans lequel les sommets sont représentés par des points du plan ou des cercles dans le plan et les arêtes sont représentées par des traits entre les sommets. Lorsqu'il existe une arête entre deux sommets x et y , on dit que x est voisin de y , et inversement. De plus, on dit que cette arête est **incidente** au sommet x et au sommet y .

Par exemple, considérons le graphe simple G défini par la paire suivante

$$G = (\{1, 2, 3, 4, 5, 6\}, \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}, \{4, 6\}\}).$$

Le graphe G a comme ensemble de sommets

$$X(G) = \{1, 2, 3, 4, 5, 6\}$$

et comme ensemble d'arêtes

$$A(G) = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}, \{4, 6\}\}.$$

Ce graphe G est représenté dans la figure 1.1.

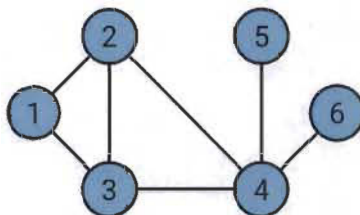


FIGURE 1.1 – Représentation d'un graphe simple G dans le plan

Définition 1.1.2 – Degré [Lab81]

Soit $G = (X, A)$, un graphe et $x \in X$, un sommet de G . Le degré de x dans G , dénoté $d(x)$, est le nombre d'arêtes de G incidentes à x , soit

$$d(x) = |\{a \in A : x \in a\}| = |\{y \in X : \{x, y\} \in A\}|.$$

Le degré d'un sommet est le nombre de voisins de ce sommet. En particulier, lorsqu'un sommet est de degré 1, on dit que le sommet est une feuille du graphe.

Un graphe dont tous les sommets sont de degré d est appelé un graphe régulier de degré d .

On note $n_i(G)$ le nombre de sommets de degré i dans le graphe G et $n(G)$ le nombre de sommets dans G .

Dans le reste du mémoire les sommets d'un graphe sont colorés selon leur degré en utilisant le code de couleurs de la figure 1.2. Le noir est utilisé pour les sommets de degré 0 et pour les sommets dont le degré est supérieur à 6.



FIGURE 1.2 – Code de couleur pour les degrés des sommets dans un graphe

Définition 1.1.3 – Chaîne [Lab81]

Dans un graphe simple $G = (X, A)$, un **chaîne simple** de longueur m est une séquence finie de sommets x_0, x_1, \dots, x_m tous distincts (sauf peut-être x_0 et x_m) telle qu'il existe une arête entre x_i et x_{i+1} pour chaque i entre 0 et m .

Lorsque $x_0 = x_m$ et que $m > 0$ on appelle la chaîne un **cycle** de longueur m de G .

Le nombre de cycles dans un graphe est intimement lié à sa structure. En général, plus un graphe a de cycles, plus ses sommets sont connectés entre eux.

Définition 1.1.4 – Graphe connexe [Lab81]

Un graphe G est **connexe** si pour chaque paire de sommets x, y de G il existe une chaîne simple de x à y .

C'est-à-dire qu'un graphe est connexe s'il est formé d'un seul bloc, que l'on ne peut pas le séparer en deux sans retirer une arête. On étudie presque uniquement les graphes connexes puisqu'un graphe non-connexe peut être considéré comme l'union disjointe de ses parties connexes.

Définition 1.1.5 – Arbre [Lab81]

Un **arbre** est un graphe simple connexe et sans cycle.

Ces concepts sont illustrés dans la figure 1.3, en (a) on a un graphe avec un cycle mis en évidence et en (b) on a un graphe sans cycle qui est donc un arbre. Dans ces deux figures, les sommets sont étiquetés avec leur degré.

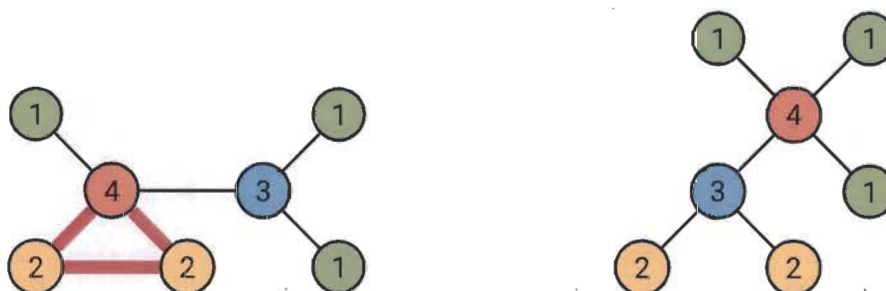


FIGURE 1.3 – Exemple de cycle dans un graphe et d'un graphe qui est un arbre

Définition 1.1.6 – Sous-graphe induit [Lab81]

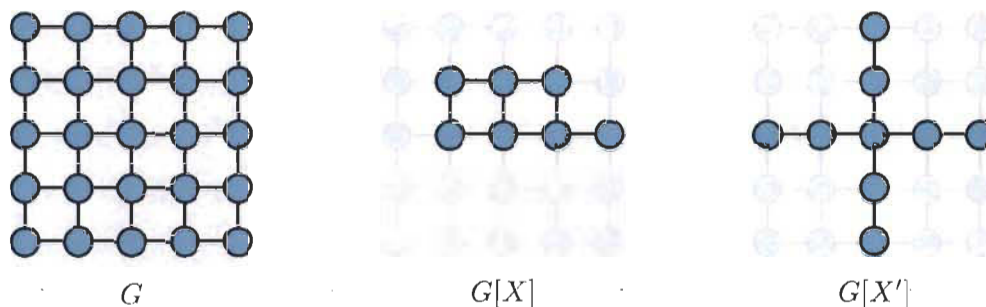
Soit $G = (X, A)$, un graphe simple, et $Y \subseteq X$ un sous-ensemble des sommets de G . Le sous-graphe induit par Y , dénoté $G[Y]$, est défini par

$$G[Y] = (Y, A \cap \mathcal{P}_2(Y)).$$

L'ensemble de tous les sous-graphes induits de taille n dans un graphe G est noté $\mathcal{T}_G(n)$, lorsque l'on considère toutes les valeurs de n , on écrit \mathcal{T}_G .

C'est-à-dire que pour construire le sous-graphe induit à partir d'un ensemble de sommets Y , on prend les sommets Y et toutes les arêtes entre des éléments de Y qui sont dans A . Un intérêt particulier est porté aux sous-graphes induits qui sont des arbres. Ces sous-graphes induits sont naturellement appelés des sous-arbres induits.

Considérons par exemple le graphe G dont les sommets sont des points du plan à coordonnées entières entre 0 et 4 et dont les arêtes sont entre les sommets à distance euclidienne 1. Ce graphe ainsi que deux sous-graphes induits pour des ensembles de sommets X et X' , dont un est un sous-arbre induit, sont illustrés à la figure 1.4.

FIGURE 1.4 – Un sous-graphe induit et un sous-arbre induit dans un graphe G

1.1.2 Automorphismes de graphe

Les sommets d'un graphe sont par définition étiquetés. C'est-à-dire que deux graphes ayant la même structure sont différents si leurs sommets sont étiquetés différemment. Les isomorphismes de graphes nous permettent de détecter lorsque deux graphes ont la même structure, mais des étiquettes différentes.

Définition 1.1.7 – Isomorphisme de graphe

Soit $G_1 = (X_1, A_1)$ et $G_2 = (X_2, A_2)$, deux graphes simples. Une fonction bijective $\phi : X_1 \rightarrow X_2$ est un **isomorphisme** de G_1 vers G_2 si elle préserve les arêtes. C'est-à-dire

$$A_2 = \{ \{ \phi(x), \phi(y) \} : \{x, y\} \in A_1 \}.$$

S'il existe un isomorphisme ϕ entre deux graphes G_1 et G_2 , on dit que G_1 et G_2 sont **isomorphes**. Plus intuitivement, deux graphes sont isomorphes s'ils ont la même structure. En particulier, un graphe est toujours isomorphe à lui-même avec l'isomorphisme identité, c'est-à-dire la fonction envoyant chaque sommet sur lui-même.

Un exemple de deux graphes isomorphes est donné dans la figure 1.5. Il n'est pas évident que les graphes G_1 et G_2 sont isomorphes, mais le déplacement ϕ des sommets de G_1

illustre bien l'isomorphisme.

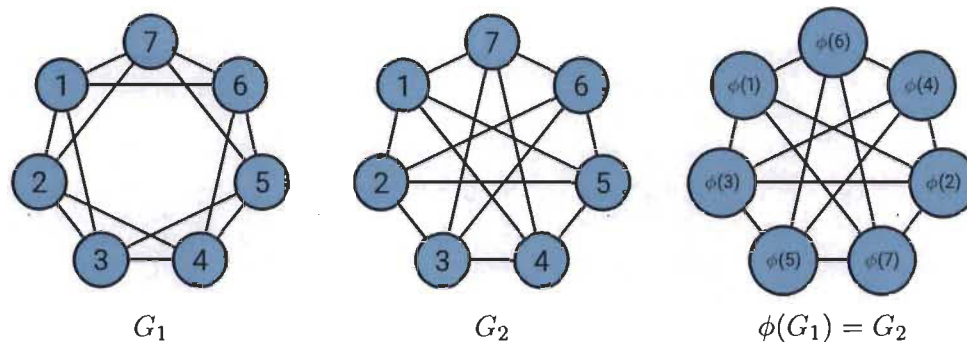


FIGURE 1.5 – Exemple d'un isomorphisme entre deux graphes

Bien que tout graphe soit isomorphe à lui-même, certains isomorphismes ne sont pas triviaux et sont donc plus intéressants.

Définition 1.1.8 – Automorphisme de graphe

Soit $G = (X, A)$ un graphe simple. Une fonction bijective $\phi : X \rightarrow X$ est un **automorphisme** sur G si elle préserve les arêtes. C'est-à-dire

$$A = \{\{\phi(x), \phi(y)\} : \{x, y\} \in A\}.$$

Comme les automorphismes préservent la structure d'un graphe, ils sont utilisés pour identifier ses symétries. Dans la figure 1.6, un automorphisme envoie les sommets 1, 2, 5, 6 les uns sur les autres et le sommet 3 sur le sommet 4. On en déduit donc que les sommets 1, 2, 5, 6 sont équivalents entre eux, tout comme les sommets 3, 4.



FIGURE 1.6 – Exemple d'un automorphisme de graphe

Proposition 1.1.1

Soit $G = (X, A)$, un graphe simple. L'ensemble des automorphismes de G forme un groupe sous la composition de fonctions.

Démonstration. Ce résultat est direct puisqu'un automorphisme est une bijection et que l'ensemble des bijections sur un ensemble forme un groupe. \square

C'est-à-dire qu'il est possible d'appliquer successivement des automorphismes à un graphe G en conservant sa structure. Cette propriété est utile dans le cas des graphes infinis et dans l'étude des polyformes et des polycubes.

1.1.3 Graphes infinis

Pour étudier les polyformes et les polycubes, on utilise une famille particulière de graphes. Comme un polyforme peut être arbitrairement grand, si l'on veut associer chaque sommet d'un graphe à une cellule d'un polyforme, la taille du graphe ne doit pas être limitée.

Définition 1.1.9 – Graphe infini

Un **graphe infini** est un graphe $G = (X, A)$ dont l'ensemble X des sommets est infini. L'ensemble A est alors défini de façon générale en compréhension, plutôt qu'en faisant la liste de ses éléments.

Les quatre graphes auxquels on s'intéresse en particulier sont des graphes infinis. Ce sont les grilles carrée, triangulaire, hexagonale et cubique.

Pour deux points p_1, p_2 du plan, la distance entre p_1 et p_2 , notée $\text{dist}(p_1, p_2)$ est définie par

la distance euclidienne, qui est

$$\begin{aligned} p_1 &= (x_1, y_1), \\ p_2 &= (x_2, y_2), \\ \text{dist}(p_1, p_2) &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \end{aligned}$$

Définition 1.1.10 – Grille carrée

La grille carrée est le graphe infini $G_{\square} = (X, A)$ où l'ensemble des sommets X est l'ensemble des paires de nombres entiers dans le plan et les arêtes A sont entre des sommets à distance euclidienne 1,

$$\begin{aligned} X &= \{(x, y) : x, y \in \mathbb{Z}\}, \\ A &= \left\{ \{(x_1, y_1), (x_2, y_2)\} : \begin{array}{l} \text{dist}((x_1, y_1), (x_2, y_2)) = 1, \\ (x_1, y_1), (x_2, y_2) \in X \end{array} \right\}. \end{aligned}$$

La grille carrée est un graphe régulier de degré 4, c'est-à-dire que tous les sommets de G_{\square} sont de degré 4. Le cycle le plus court dans G_{\square} est de longueur 4. Une partie de la grille carrée est illustrée dans la figure 1.7.

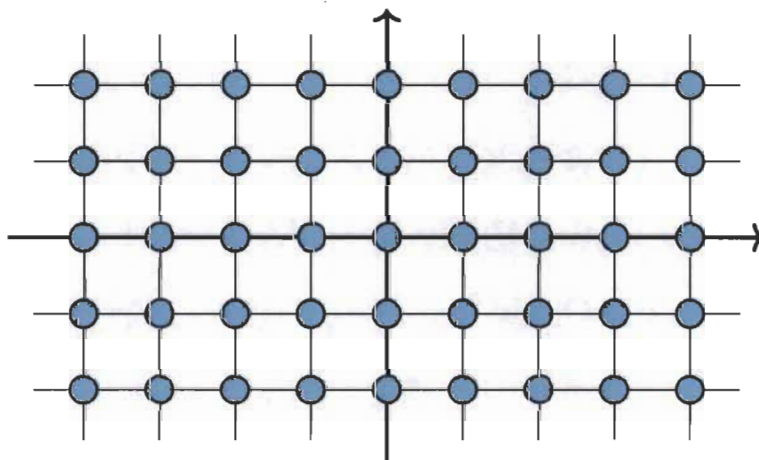


FIGURE 1.7 – Une partie de la grille carrée

Définition 1.1.11 – Grille triangulaire

La grille triangulaire est le graphe infini $G_\Delta = (X, A)$ où l'ensemble des sommets X est l'ensemble des paires de nombres réels $(x + 0.5y, \sqrt{3}/2y)$ dans le plan pour tout nombre entier x et y et dont les arêtes sont entre des sommets à distance euclidienne 1,

$$X = \left\{ \left(x + 0.5 \cdot y, \frac{\sqrt{3}}{2} \cdot y \right) : x, y \in \mathbb{Z} \right\},$$

$$A = \left\{ \{(x_1, y_1), (x_2, y_2)\} : \begin{array}{l} \text{dist}((x_1, y_1), (x_2, y_2)) = 1, \\ (x_1, y_1), (x_2, y_2) \in X \end{array} \right\}.$$

La grille triangulaire est un graphe régulier de degré 6, et son plus petit cycle est de longueur 3. Une partie de la grille triangulaire est illustrée à la figure 1.8.

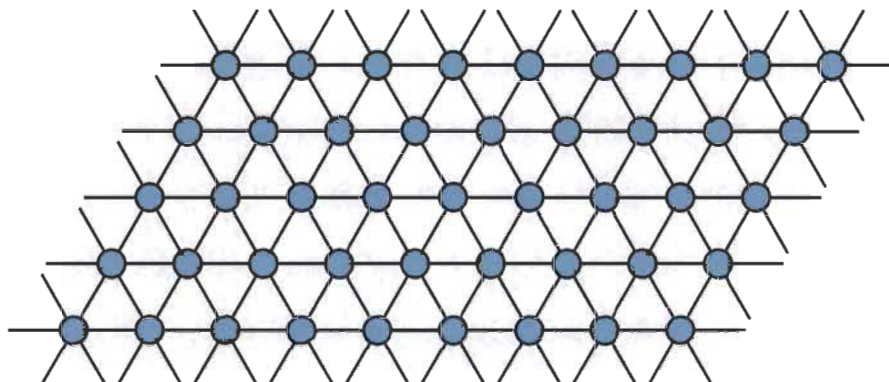


FIGURE 1.8 – Une partie de la grille triangulaire

Définition 1.1.12 – Grille hexagonale

La grille hexagonale est le graphe infini $G_O = (X, A)$ où l'ensemble des sommets X est l'ensemble des paires de nombres réels $(3x + z, \sqrt{3}/2x + \sqrt{3}y)$ dans le plan pour tous nombres entiers x et y et pour $z \in \{0, 1\}$ et l'ensemble des arêtes A est formé des paires de sommets à distance euclidienne 1,

$$X = \left\{ \left(3 \cdot x + z, \frac{\sqrt{3}}{2} \cdot x + \sqrt{3} \cdot y \right) : x, y \in \mathbb{Z}, z \in \{0, 1\} \right\},$$

$$A = \left\{ \{(x_1, y_1), (x_2, y_2)\} : \begin{array}{l} \text{dist}((x_1, y_1), (x_2, y_2)) = 1, \\ (x_1, y_1), (x_2, y_2) \in X \end{array} \right\}.$$

La grille hexagonale diffère des autres grilles par le fait qu'elle contient deux types de sommets, pour $z = 0$ et $z = 1$. La grille hexagonale est un graphe régulier de degré 3 dont le plus petit cycle est de longueur 6. Une partie de la grille hexagonale est illustrée à la figure 1.9.

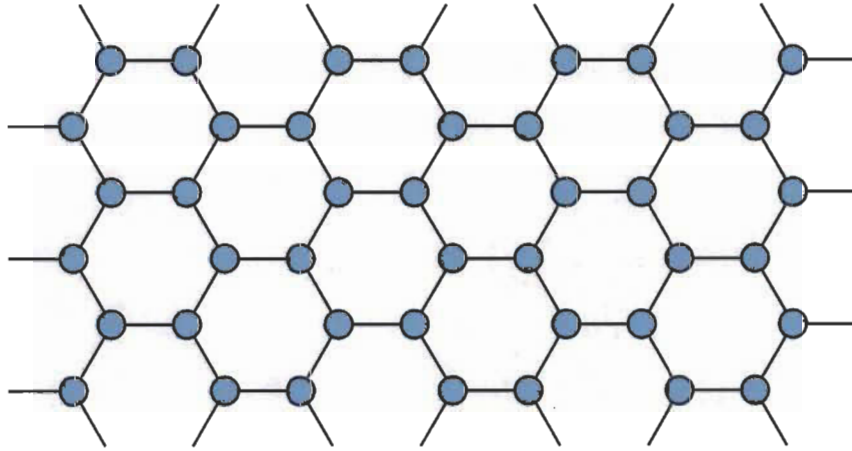


FIGURE 1.9 – Une partie de la grille hexagonale

On généralise la distance euclidienne à la dimension trois pour deux points p_1, p_2 ,

$$p_1 = (x_1, y_1, z_1),$$

$$p_2 = (x_2, y_2, z_2),$$

par

$$\text{dist}(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}.$$

Définition 1.1.13 – Grille cubique

La grille cubique est le graphe infini $G_{\square} = (X, A)$ où l'ensemble des sommets X est l'ensemble des triplets de nombres entiers dans l'espace et les arêtes A sont entre des sommets à distance euclidienne 1,

$$X = \{(x, y, z) : x, y, z \in \mathbb{Z}\},$$

$$A = \left\{ \{(x_1, y_1, z_1), (x_2, y_2, z_2)\} : \begin{array}{l} \text{dist}((x_1, y_1, z_1), (x_2, y_2, z_2)) = 1, \\ (x_1, y_1, z_1), (x_2, y_2, z_2) \in X \end{array} \right\}.$$

La grille cubique est un graphe régulier de degré 6 et son cycle le plus court est de longueur 4. Une partie de la grille cubique est illustrée dans la figure 1.10.

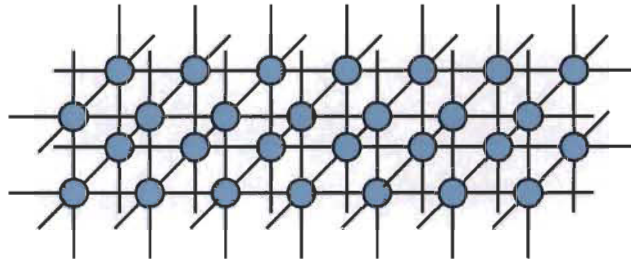


FIGURE 1.10 – Une partie de la grille cubique

1.2 Polyominos

L'étude formelle des polyominos date du milieu du 20e siècle. C'est Solomon Golomb qui a utilisé le terme pour la première fois dans le livre *Polyominos* [Gol65]. Dans cet ouvrage, il définit les polyominos et présente certains problèmes associés. Il étudie entre autres le problème de pavage par des polyominos. Ce problème demande si, étant donné une région R du plan et un ensemble \mathcal{P} de polyominos, il est possible de paver la région R avec des

polyominos de \mathcal{P} . C'est un problème qui a été démontré comme étant en général dur, et parfois impossible à résoudre.

Depuis leur introduction, les polyominos ont fait quelques apparitions dans la culture populaire, en particulier dans certains jeux. Le jeu vidéo *Tetris* est probablement l'exemple le plus connu de l'utilisation des tétraminoes, qui sont des polyominos d'aire 4. Le but de ce jeu est de paver un rectangle avec des tétraminoes. Le jeu de société *Blokus* utilise aussi les polyominos dans un contexte de pavage.

Définition 1.2.1 – Polyomino

Un **polyomino** est un ensemble de carrés unitaires qui est connexe par les côtés.

Chaque carré unitaire d'un polyomino est appelé une cellule.

Le nombre de cellules dans un polyomino est appelé l'aire ou la taille du polyomino. L'aire est une statistique qui est souvent utilisée pour classer les polyominos. C'est-à-dire que l'on s'intéresse par exemple à l'ensemble des polyominos d'aire 5, qui sont appelés des pentaminos ou en général on s'intéresse aux polyominos d'aire n .

Dans un polyomino, l'adjacence se fait par les côtés des carrés unitaires. Cette relation de connexité est appelée la 4-connexité. La 8-connexité est une alternative à cette relation qui permet la connexité par les coins et qui est utilisée entre autres en traitement d'images.

Trois familles de polyominos sont généralement étudiées, dépendant des transformations que l'on se permet de leur appliquer. On distingue les polyominos fixés, un-côté et libres.

Ces différentes familles de polyominos sont définies plus formellement plus loin. Généralement, uniquement les polyominos libres et les polyominos fixés sont étudiés. Les polyominos un-côté sont une famille intermédiaire qui est peu utilisée. Pour la suite, on étudie seulement les polyomino fixés.

Les polyominos fixés sont des structures rigides que l'on ne peut ni tourner ni renverser, alors que l'on peut appliquer n'importe quelle transformation aux polyominos libres. On

dit que les polyominos fixés sont considérés à translation près et que les polyominos libres sont considérés à symétrie près. On note $\mathcal{P}_{\square}(n)$ l'ensemble des polyominos fixés d'aire n et $p_{\square}(n)$ le nombre de polyominos fixés d'aire n .

Généralement, dans les figures, on dessine les polyominos à symétrie près, c'est-à-dire que l'on dessine des polyominos libres. Comme plusieurs polyominos fixés sont représentés par le même polyomino libre, les figures sont simplifiées. Dans la figure 1.11, les polyominos libres et fixés d'aire 1, 2 et 3 sont illustrés. La simplification obtenue en ne dessinant que les polyominos libres est alors évidente.

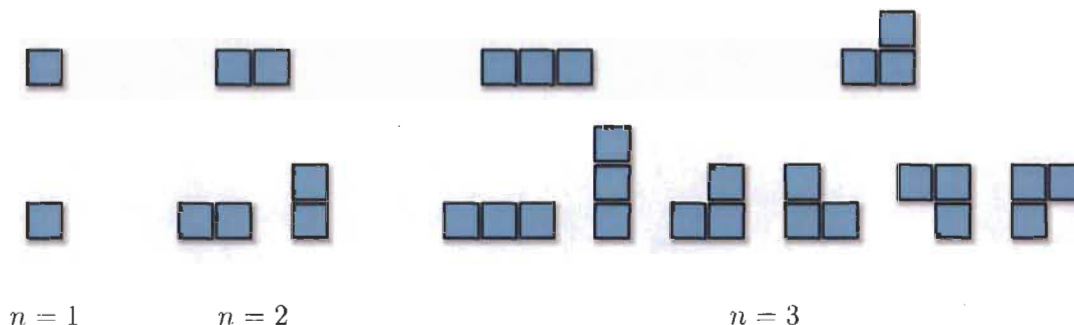


FIGURE 1.11 – Polyominos libres et fixés d'aire 1, 2 et 3

1.2.1 Graphe d'adjacence

Pour la suite, on définit un polyomino à l'aide de la théorie des graphes. Cette définition permet de simplifier les raisonnements, d'être plus rigoureux et de bénéficier du langage et des outils de la théorie des graphes.

Définition 1.2.2 – Polyomino

Un polyomino figé est un sous-graphe induit connexe de la grille carrée G_{\square} .

Par exemple, le sous-graphe induit par les sommets suivants dans G_{\square}

$$\{(2, 1), (3, 0), (3, 1), (4, -1), (4, 0), (4, 1), (4, 2)\}$$

est connexe, il correspond donc à un polyomino figé.

Certaines définitions de la théorie des graphes sont maintenant disponibles pour décrire les polyominos.

Définition 1.2.3 – Définitions associés aux graphes

Un cycle dans un polyomino est une suite de cellules dont les sommets associés forment un cycle dans le graphe d'adjacence.

Un polyomino arbre est un polyomino dont le graphe d'adjacence est un arbre.

Le degré d'une cellule d'un polyomino est le degré du sommet associé dans le graphe d'adjacence. En particulier, une feuille dans un polyomino est une cellule dont le sommet associé est de degré 1.

Les concepts de cycle et de degrés dans un polyomino sont illustrés à la figure 1.12. Dans cette figure, les sommets sont étiquetés par le degré des cellules et le cycle est mis en évidence par des lignes plus grasses.

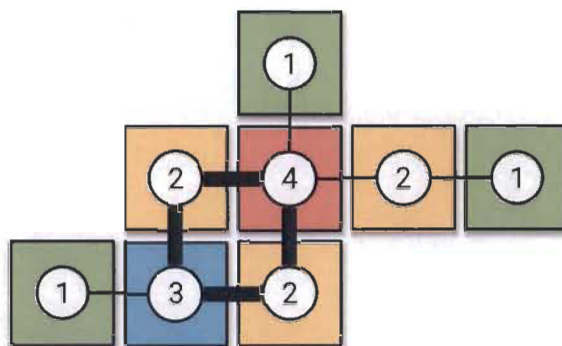


FIGURE 1.12 – Exemple de cycle et de degré dans un polyomino

Un groupe élémentaire d'automorphismes de G_{\square} est le groupe de ses translations.

Définition 1.2.4 – Translation de G_{\square}

Pour tous entiers i, j , une translation $T_{i,j}$ de la grille carrée est définie par son action sur un sommet de G_{\square} ,

$$T_{i,j}((x, y)) = (x + i, y + j).$$

L'application d'une translation sur un sous-graphe $G_{\square}[P] = (X, A)$ de G_{\square} est définie par l'application de la translation sur chacun des sommets du graphe,

$$T_{i,j}(G_{\square}[P]) = (\{T_{i,j}(x) : x \in X\}, \{\{T_{i,j}(x_1), T_{i,j}(x_2)\} : \{x_1, x_2\} \in A\}).$$

Ces translations sont obtenues en appliquant successivement les deux translations élémentaires et leur inverses, qui sont

$$T_{0,1}((x, y)) = (x, y + 1),$$

$$T_{1,0}((x, y)) = (x + 1, y).$$

C'est-à-dire que l'application d'une translation équivaut à un déplacement des sommets de G_{\square} i unités horizontalement et j unités verticalement. En particulier, la translation $T_{0,0}$ est la translation identité, qui ne déplace pas les sommets.

On constate qu'une translation est un automorphisme de G_{\square} . D'abord, une translation est une fonction bijective sur les sommets. De plus, une translation est une isométrie, c'est-à-dire que deux sommets à distance 1 avant l'application de la translation restent à distance 1 après son application.

On dit alors que deux sous-graphes induits G_1, G_2 de G_{\square} sont équivalents à translation près, noté $G_1 \equiv_T G_2$ s'il existe une translation $T_{i,j}$ telle que

$$G_1 = T_{i,j}(G_2)$$

Définition 1.2.5 – Polyominos fixés

L'ensemble des polyominos fixés est l'ensemble des sous-graphes induits connexes de G_{\square} quotienté par la relation \equiv_T .

C'est-à-dire que deux polyominos fixés P_1, P_2 sont égaux si l'on peut obtenir P_1 en translatant P_2 .

La correspondance entre les définitions 1.2.1 et 1.2.5 de polyomino se fait en associant chaque sommet du sous-graphe induit au carré unitaire dont le coin inférieur gauche est sur le sommet du sous-graphe induit. C'est-à-dire que le sommet (x, y) de la grille carrée est associé au carré unitaire dont les coordonnées des coins sont

$$(x, y), (x + 1, y), (x, y + 1), (x + 1, y + 1).$$

On appelle le **graphe d'adjacence** d'un polyomino le sous-graphe induit de G_{\square} associé au polyomino. Cette correspondance est illustrée dans la figure 1.13 entre un sous-graphe induit de G_{\square} en (a) et un polyomino en (b).

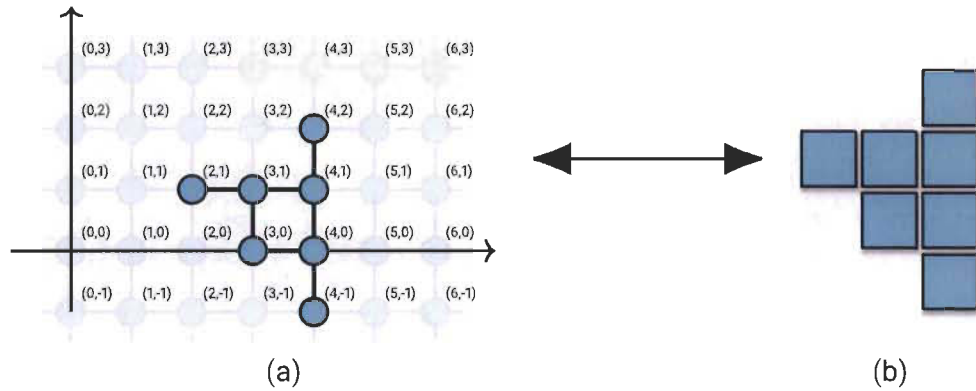


FIGURE 1.13 – La correspondance entre les polyominos et les sous-graphes induits de G_{\square}

Il est possible d'obtenir un unique représentant d'une classe de polyominos fixés en translatant un polyomino dans le premier quadrant et en minimisant ses coordonnées. Pour le sous-graphe induit $P = (X, A)$ de G_{\square} , notons x_{\min} et y_{\min} les coordonnées minimales horizontales et verticales respectivement des cellules de P ,

$$x_{\min} = \min(\{x : (x, y) \in X\}), \quad y_{\min} = \min(\{y : (x, y) \in X\}).$$

Le représentant de P à translation près, noté $T_{min}(P)$ est défini par

$$T_{min}(P) = T_{-x_{min}, -y_{min}}(P).$$

Considérons le sous-graphe induit de G_{\square} illustré dans la figure 1.14 (a), ses coordonnées minimales sont $x_{min} = 2, y_{min} = -1$, en appliquant la translation $T_{-2,1}$, on obtient donc le représentant d'une classe de polyominos figés à translation près, qui est illustré dans la figure 1.14 (b).

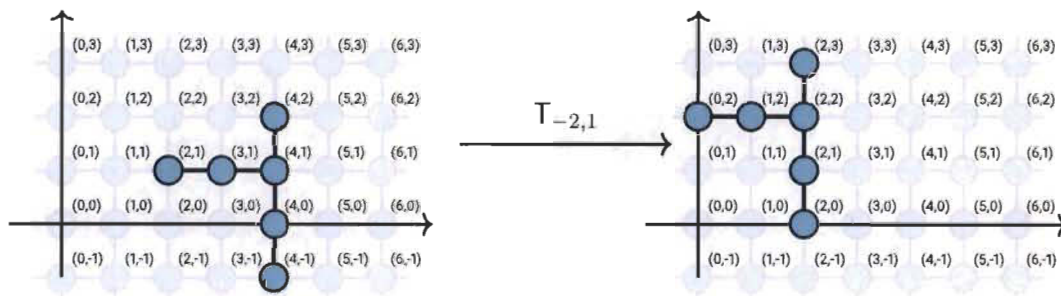


FIGURE 1.14 – Choix d'un représentant dans une classe de polyominos à translation près

Pour la suite, on considère uniquement les représentants des polyominos fixés. C'est-à-dire qu'un polyomino est un ensemble de carrés unitaires qui est connexe, dont tous les carrés sont dans le premier quadrant et dont toutes les coordonnées sont minimisées.

1.2.2 Dénombrement

Le problème principal en lien avec les polyominos est de les compter. Étant donné un nombre n , on s'intéresse au nombre de polyominos que l'on peut construire en utilisant n carrés. Deux approches sont généralement utilisées, la génération informatique et la décomposition combinatoire.

La génération informatique ne donne aucun résultat général, et est limitée à des aires relativement petites. En effet, comme le nombre de polyominos est exponentiel, la mémoire

et le temps de calcul nécessaires augmentent très rapidement. L'algorithme le plus efficace pour l'énumération des polyominos fixés est l'algorithme de Jensen [Jen01].

Cet algorithme a été utilisé pour trouver le nombre de polyominos fixés de taille 56, qui est

$$69150714562532896936574425480218.$$

D'autres algorithmes existent, comme celui de Redelmeier [Red81]. Cet algorithme a pour avantage de pouvoir générer les polyomino libres.

On connaît donc le nombre de polyominos libres de taille 45, qui est

$$2155600091107324229254415.$$

Bien que l'on ne connaisse pas le nombre exacte de polyominos de taille donnée, on peut faire des estimations asymptotiques. On cherche donc des formules qui se rapprochent du nombre de polyominos pour de grandes valeurs de n . L'estimation la plus précise obtenue dans l'article [BMRR06] est

$$\frac{0.3024 \cdot 4.0646^n}{n}.$$

Les premières valeurs du nombre de polyominos libres et fixés de taille donnée sont données dans la table 1.1. Ces valeurs sont obtenues informatiquement.

OEIS	n	1	2	3	4	5	6	7	8	9	10	11
A001168	Fixés	1	2	6	19	63	216	760	2725	9910	36446	135268
A000105	Libres	1	1	2	5	12	35	108	369	1285	4655	17073

TABLE 1.1 – Nombre de polyominos fixés et libres

La décomposition combinatoire a été jusqu'à maintenant infructueuse dans le cas général. Pour certaines familles restreintes de polyominos, il est possible d'obtenir un résultat général donnant le nombre de polyominos de cette famille.

Parmi les familles qui ont été énumérées de façon exacte, on compte les polyominos d'aire minimale dans un rectangle dans [GCN10], [GCP13] et [Pel15], les serpents partiellement dirigés dans [GPdWd18] et [dWd16], les polyominos d'aire maximale (gominos) [FGLT13] et les polyominos convexes [BGOR99].

D'autres familles ont aussi étudiées et énumérées comme les polyominos bargraphs [BMR03] et les polyominos parallélogramme [DGBV87].

1.3 Polyhexes et polytriangles

Les polyhexes et les polytriangles sont, avec les polyominos, les trois seules familles de polyformes dans le plan.

Définition 1.3.1 – Polyhexe

Un polyhexe est un ensemble d'hexagones réguliers qui est connexe par les côtés.

Définition 1.3.2 – Polytriangle

Un polytriangle est un ensemble de triangles équilatéraux qui est connexe par les côtés.

Une particularité des polytriangles est qu'ils contiennent des triangles dans deux orientations, l'un ayant subi une rotation de 180° par rapport à l'autre.

Ces deux définitions sont analogues à la définition de polyomino, c'est-à-dire que les polyhexes et les polytriangles sont des sous-ensembles connexes par les côtés d'un pavage régulier.

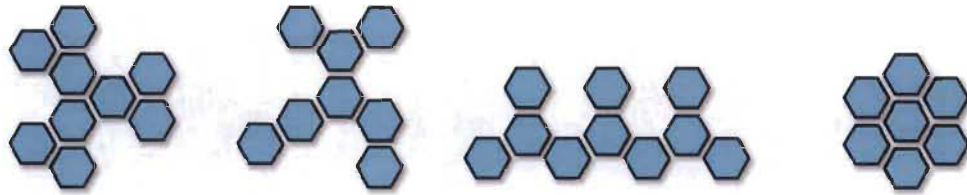


FIGURE 1.15 – Exemples de polyhexe

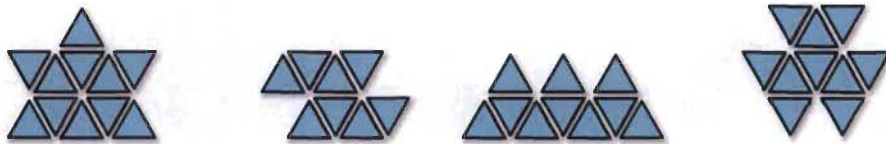


FIGURE 1.16 – Exemples de polytriangle

Chaque triangle ou hexagone d'un polytriangle ou d'un polyhexe est appelé une **cellule**. Le nombre de cellules dans un polyforme est appelé sa **taille**. La taille est une statistique souvent utilisée pour partitionner les polytriangles et les polyhexes.

Trois familles de polyformes sont généralement étudiées, dépendant des transformations que l'on se permet de leur appliquer. On distingue les polytriangles et les polyhexes **fixés**, **un-côté** et **libres**. Seulement les polyformes fixés sont définis et utilisés plus tard.

1.3.1 Graphe d'adjacence

Comme avec les polyominos, on définit plus rigoureusement les polyhexes et les polytriangles comme des sous-graphes induits de graphes infinis.

Définition 1.3.3 – Polyhexe et polytriangle

Un polyhexe figé (resp. polytriangle figé) est un sous-graphe induit connexe de la grille triangulaire (resp. hexagonale).

De cette définition, on obtient le degré d'une cellule dans un polyforme, la présence d'un cycle dans un polyforme et un polyforme arbre.

L'inversion entre les hexagones et les triangles dans la définition 1.3.3 est intentionnelle, c'est-à-dire qu'un polyhexe figé est un sous-graphe induit de la grille triangulaire et qu'un polytriangle est un sous-graphe induit de la grille hexagonale. Les grilles tiennent leur nom des formes des arêtes des graphes. Par contre, dans un sous-graphe induit, les cellules des polyformes sont associées aux sommets des graphes. Cette inversion est due au fait que la grille hexagonale est le dual de la grille triangulaire. C'est-à-dire qu'en remplaçant les faces formées par les arêtes dans la grille hexagonale par des sommets et en les reliant si elles partagent une arête, on obtient la grille triangulaire.

Définition 1.3.4 – Translations de G_O

Pour tous entiers i, j , une translation $T_{i,j}$ de la grille hexagonale est définie par son action sur un sommet :

$$T_{i,j}((x, y)) = \left(x + 3 \cdot i + 1.5 \cdot j, y + \frac{\sqrt{3}}{2} \cdot j \right).$$

Définition 1.3.5 – Translations de la grille triangulaire

Pour tous entiers i, j , une translation $T_{i,j}$ de la grille triangulaire est définie par son action sur un sommet :

$$T_{i,j}((x, y)) = \left(x + i + 0.5 \cdot j, y + \frac{\sqrt{3}}{2} \cdot j \right).$$

Ces translations sont obtenues en appliquant successivement deux translations élémentaires et leur inverse. Les groupes des translations de ces grilles sont donc des groupes abéliens libres à deux générateurs. Pour la grille triangulaire, ces translations sont

$$T_{0,1}((x, y)) = \left(x + 0.5, y + \frac{\sqrt{3}}{2} \right),$$

$$T_{1,0}((x, y)) = (x + 1, y).$$

Pour la grille hexagonale, ces translations sont

$$T_{0,1}((x, y)) = \left(x + 1.5, y + \frac{\sqrt{3}}{2} \right),$$

$$T_{1,0}((x, y)) = (x + 3, y).$$

Dans les deux cas, une des translations est purement horizontale, d'une distance de 3 unités pour la grille hexagonale et de 1.5 unités pour la grille triangulaire. La seconde translation est diagonale, combinant une translation verticale et une translation horizontale. En appliquant successivement ces deux translations, on obtient toutes les translations possibles des grilles hexagonales et triangulaires.

Deux polyhexes (resp. polytriangles) P_1, P_2 sont équivalents à translation près, noté $P_1 \equiv_T P_2$, s'il existe une translation $T_{i,j}$ de la grille triangulaire (resp. hexagonale) telle que

$$P_1 = T_{i,j}(P_2).$$

Comme pour les polyominos, on étudie que deux familles de polyhexes et de polytriangles, libres et fixés. Dans la figure 1.17, il y a 6 polytriangles fixés différents qui sont tous égaux lorsqu'ils sont considérés comme des polytriangles libres.



FIGURE 1.17 – Exemple de polytriangles fixés et libres

Définition 1.3.6 – Polyhexes et polytriangles fixés

L'ensemble des polyhexes (resp. polytriangles) fixés est l'ensemble des sous-graphes induits connexes de la grille triangulaire (resp. hexagonale) quotienté par la relation \equiv_T .

C'est-à-dire que deux polyformes fixés P_1, P_2 sont égaux si l'on peut translater P_1 pour obtenir P_2 .

L'équivalence entre les définitions utilisant les sous-graphes induits et celles utilisant formes géométriques se trouve en remplaçant chaque sommet d'un sous-graphe par un triangle ou un hexagone de façon à placer les côtés de la forme perpendiculaire aux arêtes incidentes au sommet. Cette correspondance est illustrée pour les polytriangles et la grille hexagonale à la figure 1.18. Le sous-graphe induit associé à un polyforme par cette correspondance est appelé son *graphe d'adjacence*.

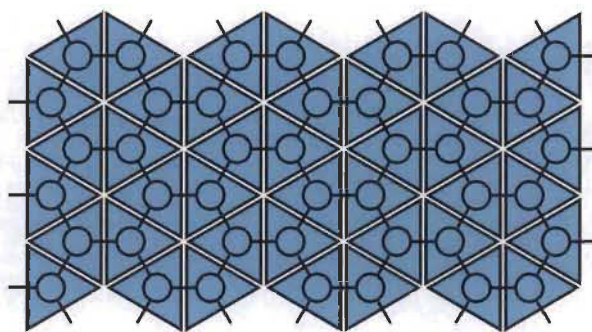


FIGURE 1.18 – Correspondance entre polytriangles et sous-graphes induits de la grille hexagonale

Afin de représenter un polyforme fixé on définit un représentant de chaque polytriangle et polyhexe à translation près. Pour ce faire, on déplace chaque polytriangle ou polyhexe dans le premier quadrant, et on minimise les coordonnées de ses sommets. Cette procédure est équivalente à celle pour les polyominos.

1.3.2 Dénombrement

Comme pour les polyominos, on s'intéresse au dénombrement des polyhexes et des polytriangles fixés et libres. Les premières valeurs du nombre de polyhexes et de polytriangles fixés et libres est donné dans le table 1.2. Comme pour les polyominos, ces valeurs sont obtenues par génération informatique. Des algorithmes adaptés de l'algorithme de Redelmeier sont utilisés pour la génération [Lun72]. On ne connaît donc aucun résultat général sur l'énumération des polyhexes et des polytriangles à n cellules.

OEIS	n	1	2	3	4	5	6	7	8	9
A001207	Polyhexes fixés	1	3	11	44	186	814	3652	16689	77359
A001420	Polytriangles fixés	2	3	6	14	36	94	250	675	1838
A002212	Polyhexes libres	1	1	3	10	36	137	543	2219	9285
A000577	Polytriangles libres	1	1	1	3	4	12	24	66	160

TABLE 1.2 – Nombre de polyhexes et polytriangles fixés et libres

1.4 Polycubes

Les polycubes sont les équivalents tridimensionnels des polyominoes. Ils sont une généralisation des polyformes dont les cellules sont des cubes. Le passage de la dimension 2 à la dimension 3 complexifie significativement les résultats.

Définition 1.4.1 – Polycube

Un polycube est un ensemble de cubes unitaires qui est connexe par les faces.

Chaque cube dans un polycube est appelé une cellule. Le nombre de cellules dans un polycube est appelé sa taille. Des polycubes de tailles 7 et 10 sont illustrés dans la figure 1.19.

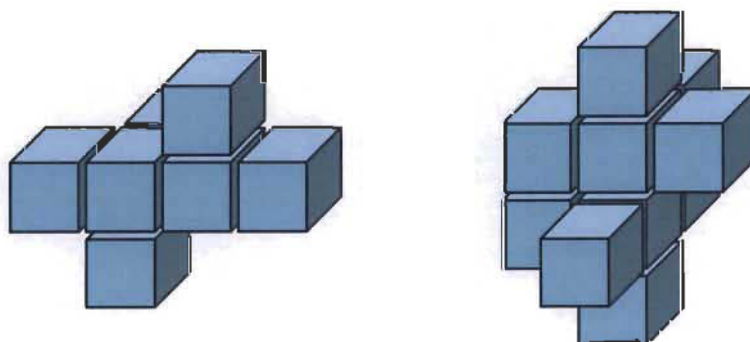


FIGURE 1.19 – Polycubes de taille 7 et 10

Les polycubes partagent plusieurs propriétés avec les polyominos. On peut donc étudier des problèmes similaires à ceux sur les polyominos. Par exemple, étant donné un ensemble de polycubes, est-il possible de paver une région de l'espace de dimension trois avec des éléments de cet ensemble ?

Similairement aux polyominos, il serait possible de généraliser les polycubes en changeant la relation d'adjacence. Un polycube doit être connexe par les faces, mais on pourrait permettre la connexité par les arêtes ou encore par les sommets.

Comme pour les polyformes, on distingue trois familles de polycubes. Les polycubes fixés sont des polycubes à translation près, les polycubes un-côté préservent l'orientation de l'espace et les polycubes libres sont des polycubes à symétrie près. Comme dans les sections précédentes, on illustre les polycubes à symétrie près dans les figures pour réduire leur taille, mais on les étudie à translation près. Dans la figure 1.20, 12 polycubes fixés différents sont illustrés qui sont tous égaux à symétrie près.

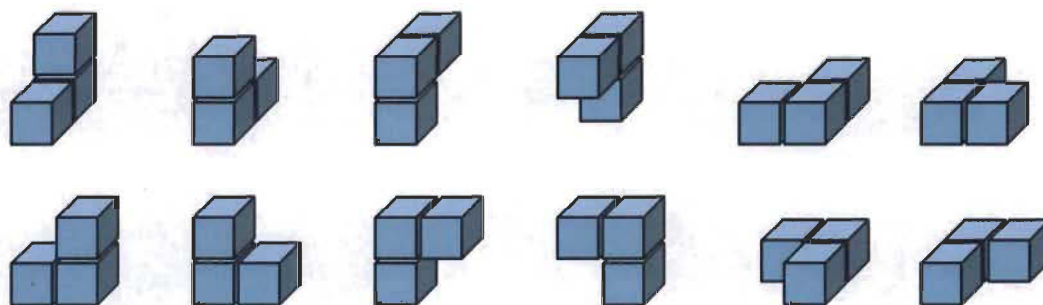


FIGURE 1.20 – Polycubes fixés équivalents à symétrie près

1.4.1 Graphe d'adjacence

Afin de formaliser les définitions et de bénéficier des résultats de la théorie des graphes, on utilise une définition de polycube basée sur les sous-graphes induits.

Définition 1.4.2 – Polycube (graphe)

Un polycube figé est un sous-graphe induit connexe de la grille cubique G_{\square} .

Comme précédemment, deux polycubes figés ayant la même structure peuvent être situés à différents endroits dans la grille cubique, on définit donc les translations de G_{\square} afin d'identifier ces structures.

Définition 1.4.3 – Translations tridimensionnelles

Pour tout entiers i, j, k , une translation $T_{i,j,k}$ de la grille cubique est définie par son action sur un sommet de G_{\square} ,

$$T_{i,j,k}((x, y, z)) = (x + i, y + j, z + k).$$

Comme pour le cas des polyformes, on définit l'application d'une translation sur un sous-graphe de la grille cubique P en l'appliquant sur chaque sommet du sous-graphe.

$$T_{i,j,k}(P) = \{T_{i,j,k}(c) : c \in P\}.$$

Ces translations sont obtenues en appliquant successivement trois translations élémentaires qui sont

$$T_{0,0,1}((x, y, z)) = (x, y, z + 1),$$

$$T_{0,1,0}((x, y, z)) = (x, y + 1, z),$$

$$T_{1,0,0}((x, y, z)) = (x + 1, y, z).$$

Ces translations sont toutes des déplacements de 1 unité dans la direction d'un axe de coordonnées. Le groupe des translations de la grille cubique forme donc un groupe abélien libre à trois générateurs.

On dit que deux sous-graphes induits P_1, P_2 de G_{\square} sont équivalents à translation près et on écrit $P_1 \equiv_T P_2$, s'il existe une translation $T_{i,j,k}$ telle que

$$P_1 = T_{i,j,k}(P_2).$$

Définition 1.4.4 – Polycubes fixés

L'ensemble des polycubes fixés est l'ensemble des sous-graphes induits connexes de la grille cubique quotienté par la relation \equiv_T .

C'est-à-dire que deux polycube fixés P_1, P_2 sont égaux si on peut translater P_1 pour obtenir P_2 .

Pour lier les définitions 1.4.1 et 1.4.4, on définit une correspondance entre un sommet d'un sous-graphe induit et une cellule d'un polycube. On associe au sommet (i, j, k) de $G_{\mathbf{a}}$ le cube unitaire dont les huit sommets sont

$$\begin{array}{cccc} (i, j, k), & (i, j, k + 1), & (i, j + 1, k), & (i, j + 1, k + 1) \\ (i + 1, j, k), & (i + 1, j, k + 1), & (i + 1, j + 1, k), & (i + 1, j + 1, k + 1). \end{array}$$

Par exemple, le sommet $(0, 0, 0)$ de la grille cubique correspond au cube dont deux coins opposés sont $(0, 0, 0)$ et $(1, 1, 1)$. Cette correspondance est illustrée dans la figure 1.21.

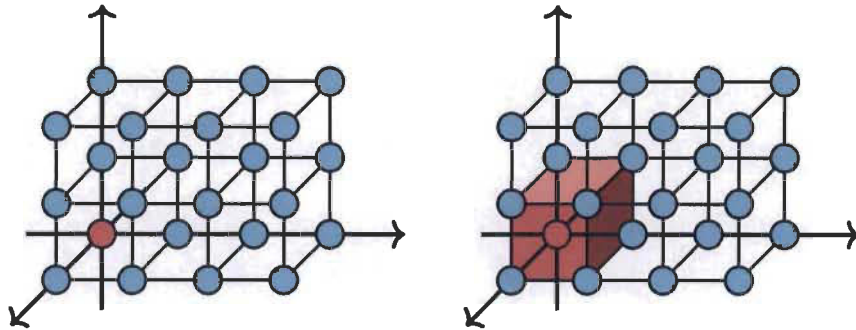


FIGURE 1.21 – Correspondance entre un sommet de la grille cubique et une cellule dans un polycube

Le sous-graphe induit associé à un polycube par cette correspondance est appelé son graphe d'adjacence. De cette définition, on obtient les concepts de cycle et de degré d'une cellule dans un polycube.

Afin d'obtenir un représentant unique pour chaque polycube fixé dans \mathbb{Z}^3 , on translate les

polycubes dans le premier octant et on minimise ses coordonnées. On définit d'abord les constantes suivantes pour un polycube P

$$x_{\min} = \min(\{x : (x, y, z) \in P\}),$$

$$y_{\min} = \min(\{y : (x, y, z) \in P\}),$$

$$z_{\min} = \min(\{z : (x, y, z) \in P\}).$$

Le représentant de P à translation près, noté T_{\min} est défini par

$$T_{\min}(P) = T_{-x_{\min}, -y_{\min}, -z_{\min}}(P).$$

1.4.2 Dénombrement

Comme pour les polyominos, on s'intéresse au dénombrement des polycubes libres et fixés selon la taille. Les premières valeurs du nombre de polycubes libres et fixés de taille n sont données dans le table 1.1. Comme pour les polyominos, ces valeurs ont été obtenues par génération informatique. Encore une fois, l'algorithme de Redelmeier peut être modifié pour générer des polycubes [Red81]. On ne connaît aucun résultat général sur l'énumération des polycubes selon le volume.

OEIS	n	1	2	3	4	5	6	7	8	9
A000162	Polycubes libres	1	1	2	8	29	166	1023	6922	48311
A001931	Polycubes fixés	1	3	15	86	534	3481	23502	162913	1152870

TABLE 1.3 – Nombre de polycubes fixés et libres

Chapitre 2

Sous-arbres induits pleinement feuillus

L'arbre est dans ses feuilles Marilon Marilé,

L'arbre est dans ses feuilles Marilon Don Dé.

— Zachary Richard

Les concepts utiles à l'étude des arbres pleinement feuillus sont introduits dans ce chapitre. Les définitions sont formulées de façon générale afin qu'elles soient applicables dans les chapitres subséquents. Ce chapitre se veut donc une introduction à l'étude des arbres pleinement feuillus. En général, les définitions utilisées ici n'existent pas dans la littérature et sont spécifiques à l'étude des arbres pleinement feuillus.

À la section 2.1, on définit les fonctions feuille, qui sont utilisées pour caractériser les arbres pleinement feuillus. Les sections 2.2 et 2.3 décrivent les objets utilisés pour trouver les fonctions feuille et les méthodes de preuve utilisant ces objets. La section 2.4 introduit les arbres saturés, qui sont des arbres pleinement feuillus ayant des caractéristiques supplémentaires restreignant leur structure.

2.1 Fonction feuille

L'objet central dans l'étude du nombre maximal de feuilles dans les sous-arbres induits est la fonction feuille d'un graphe connexe qui a été introduit dans l'article [BdG17]. Cette fonction identifie le nombre maximal de feuilles des sous-arbres induits.

Rappelons qu'un arbre est un graphe sans cycle et qu'une feuille dans un arbre est un sommet de degré 1.

Il est à noter que si deux sous-arbres induits sont isomorphes, ils ont nécessairement le même nombre de feuilles. C'est-à-dire que le nombre de feuilles est uniquement lié à la structure de graphe. En particulier, dans le cas des polyformes et des polycubes, l'application d'une translation ou d'une autre symétrie du graphe ne modifie jamais le nombre de feuilles d'un sous-arbre induit.

Définition 2.1.1 – Fonction feuille

Soit G un graphe et $\mathcal{T}_G(n)$ l'ensemble des sous-arbres induits à n sommets dans G . La fonction feuille de G , notée L_G , est une fonction dont le domaine et l'image est l'ensemble des nombres naturels supérieurs à 1. La fonction L_G est définie par

$$L_G(n) = \max\{n_1(T) : T \in \mathcal{T}_G(n)\}, \forall n \geq 2.$$

Lorsque $\mathcal{T}_G(n) = \emptyset$ pour un certain n , on pose $L_G(n) = \max(\emptyset) = -\infty$. Un sous-arbre induit T de taille n ayant $L_G(n)$ feuilles est dit **pleinement feuillu**.

Les arbres pleinement feuillus ont été introduits dans l'article [BdG17] en 2017. Dans cet article, le nombre maximal de feuilles dans les polyominos arbre a d'abord été étudié. Une bijection entre des familles de polyominos arbre donnant des informations sur les polyominos arbre pleinement feuillus a aussi été obtenue.

Le problème a été généralisé aux polytriangles et aux polyhexes. Ainsi, une énumération exhaustive des polytriangles arbre pleinement feuillus et des polyhexes arbre pleinement

feuillus a été complétée dans l'article [BdG18a].

La spécialisation du problème des sous-arbres induits pleinement feuillus aux polycubes est non triviale et permet d'obtenir des structures inattendues et intéressantes dans [BdG17].

Le problème des sous-arbres induits pleinement feuillus a ensuite été généralisé aux graphes quelconques dans l'article [BdG⁺18c] avec une étude approfondie du nombre maximal de feuilles dans les chenilles, qui sont une famille particulière de graphes, dans l'article [BdG⁺18b]. En général, la recherche du nombre maximal de feuilles dans les sous-arbres induits d'un graphe est appelé le problème des sous-arbres induits pleinement feuillus, abrégé par FLIS pour *Fully Leafed Induced Subtree*.

Il existe un problème similaire au problème des sous-arbres induits pleinement feuillus en informatique théorique, le problème du nombre maximal de feuilles d'un arbre couvrant dans un graphe. Ce problème étudié dans l'article [GMM94] s'intéresse aussi au nombre de feuilles dans les sous-arbres d'un graphe. La différence est que ce problème s'intéresse aux arbres couvrants, plutôt qu'aux sous-arbres induits. Un arbre couvrant dans un graphe doit contenir tous les sommets du graphe, mais pas nécessairement toutes les arêtes.

Les sous-arbres induits d'un graphe, auxquels on s'intéresse dans ce chapitre, sont introduits dans la définition 1.1.6 du chapitre précédent. Ils sont obtenus en choisissant des sommets d'un graphe et en prenant toutes les arêtes du graphe reliant chaque paire de sommets de sorte que l'on obtienne un arbre. L'utilisation des sous-arbres induits introduit des contraintes qui donnent une structure intéressante aux sous-arbres induits pleinement feuillus.

Certains résultats et certaines définitions sur les sous-arbres induits pleinement feuillus sont applicables pour tous les graphes. On utilise ces résultats dans tous les chapitres qui suivent. Il est donc utile de définir ces concepts dans le cas général. On peut alors montrer des résultats une seule fois plutôt que de les répéter pour chaque famille de graphe étudié.

Dans ce chapitre, on utilise les concepts de théorie des graphes introduits à la section 1.1. Les polyformes et polycubes introduits aux sections 1.2, 1.3 et 1.4 sont aussi utilisés.

La greffe d'arbres introduite dans [BdG17] est une opération utilisée abondamment pour trouver les arbres pleinement feuillus. On utilise cette opération abondamment dans ce mémoire.

Pour se familiariser avec les fonction feuilles, considérons le graphe G illustré à la figure 2.1 (a). Des sous-arbres induits pleinement feuillus de G sont illustrés à la figure 2.1 (b) à (e) en ordre croissant de taille allant de 2 à 5. Ces sous-arbres induits pleinement feuillus ne sont pas nécessairement uniques. Il n'y a aucun sous-arbre induit à 6 sommets. Les valeurs de la fonction feuille de G sont données dans la table 2.1.

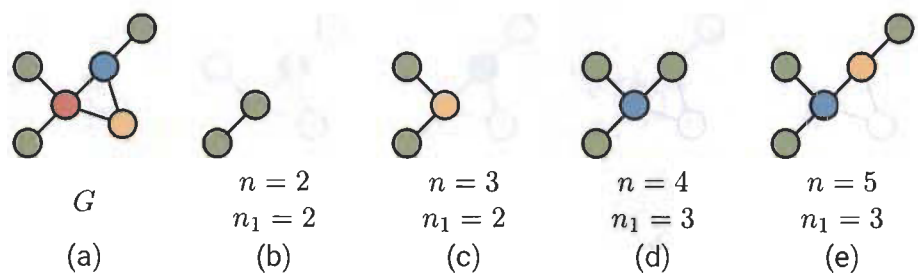
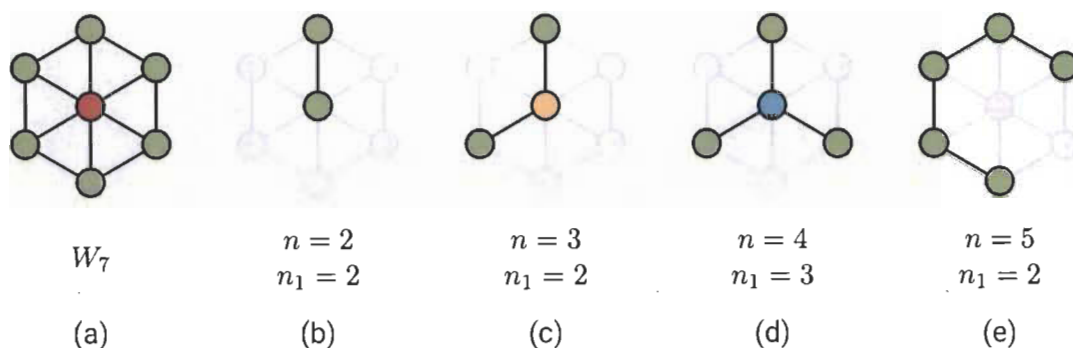


FIGURE 2.1 – Sous-arbres induits pleinement feuillus d'un graphe

n	2	3	4	5	6
$L_G(n)$	2	2	3	3	$-\infty$

TABLE 2.1 – Valeurs de la fonction feuille du graphe G de la figure 2.1

À titre d'exemple supplémentaire, on cherche la fonction feuille du graphe appelée la roue d'ordre 7, noté W_7 , illustrée à la figure 2.2 (a). Le graphe W_7 est formé d'un sommet central relié à 6 sommets externes reliés entre eux. Des sous-arbres induits pleinement feuillus d'aire 2 à 5 sont illustrés à la figure 2.2 (b) à (e). On en déduit les valeurs de sa fonction feuille, qui sont données dans la table 2.2. Il n'existe pas de sous-graphe induit de W_7 de taille 6 et 7 qui soit un arbre.

FIGURE 2.2 – Sous-arbres induits pleinement feuillus du graphe W_7

n	2	3	4	5	6	7
$L_{W_7}(n)$	2	2	3	2	$-\infty$	$-\infty$

TABLE 2.2 – Valeurs de la fonction feuille du graphe W_7 de la figure 2.2

C'est-à-dire que pour un nombre entier n , la valeur de $L_G(n)$ donne le nombre maximal de feuilles que peut avoir un sous-arbre induit à n sommets dans G . Un sous-arbre induit pleinement feuillu a donc le nombre maximal de feuilles qu'un arbre peut avoir pour sa taille.

En général, aucune méthode efficace n'est connue pour trouver les valeurs de $L_G(n)$ pour un graphe G quelconque.

Certaines contraintes permettent de caractériser les valeurs que peuvent prendre la fonction feuille d'un graphe quelconque. En particulier, les contraintes de la proposition 2.1.1 sont utiles pour conjecturer les valeurs de la fonction feuille d'un graphe donné.

Proposition 2.1.1

Soit G un graphe ayant au moins deux sommets et une arête. Soit n et k , deux entiers positifs tels que $n > 2$ et $k < n + 1$. On a

$$L_G(2) = 2, \quad (2.1)$$

$$L_G(n) \leq n - 1, \quad (2.2)$$

$$L_G(n + k) \leq L_G(n) + k. \quad (2.3)$$

Démonstration. L'équation 2.1 s'obtient en choisissant une arête $\{x, y\}$ quelconque dans G . Le graphe induit par le choix des sommets $\{x, y\}$ est un arbre dont les deux sommets sont des feuilles, voir figure 2.3 (a). Comme on ne peut pas avoir plus de feuilles que de sommets, $L_G(2) = 2$.

Pour l'équation 2.2, considérons un arbre quelconque ayant le plus de feuilles, sans autre contrainte. Cet arbre a $n - 1$ feuilles reliées à un sommet de degré $n - 1$, donnant n sommets au total, tel qu'illustré dans la figure 2.3 (b). On ne peut pas avoir un graphe composé de n feuilles toutes reliées entre elles. On a donc $L_G(n) \leq n - 1$.

Finalement, pour l'équation 2.3 on montre uniquement le cas $k = 1$, les autres cas suivent naturellement par induction. Supposons qu'il existe un sous-arbre induit T de G de taille $n + 1$ ayant $L_G(n + 1)$ feuilles. En retirant une feuille à T , on obtient un arbre T' de taille n ayant $L_G(n + 1) - 1$ ou $L_G(n + 1)$ feuilles, voir figure 2.3 (c). On a donc

$$n_1(T') \geq L_G(n + 1) - 1.$$

Comme T' est un sous-arbre induit de G , on a

$$\begin{aligned} L_G(n + 1) - 1 &\leq n_1(T') \leq L_G(n), \\ \Rightarrow \quad L_G(n + 1) &\leq L_G(n) + 1. \end{aligned}$$

En appliquant ce raisonnement k fois, on obtient le résultat désiré. □

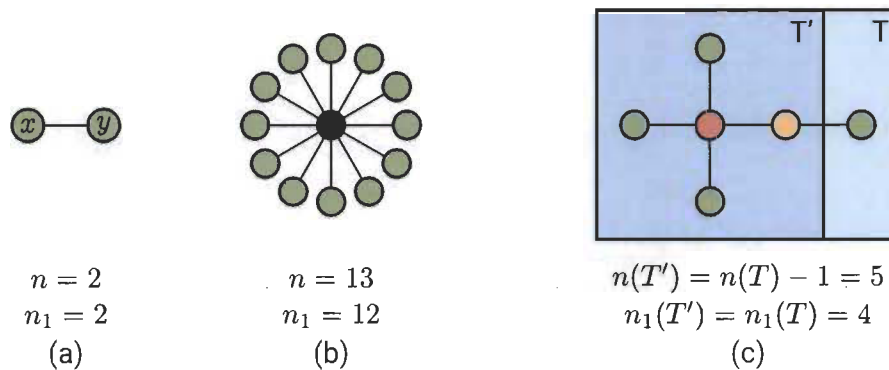


FIGURE 2.3 – Illustration des démonstrations de la proposition 2.1.1

Définition 2.1.2 – Fonction feuille potentielle

Une fonction f est une fonction feuille potentielle si son domaine et son image est l'ensemble

$$\{n : n \in \mathbb{N} \text{ et } 2 \leq n \leq m\}$$

pour un certain entier m , possiblement infini, et si elle respecte toutes les propriétés de la proposition 2.1.1.

C'est-à-dire qu'une fonction feuille potentielle respecte toutes les propriétés que l'on attend d'une fonction feuille. En d'autres mots, si une fonction n'est pas une fonction feuille potentielle, alors elle n'est pas une fonction feuille.

La figure 2.4 illustre la région du plan dans laquelle le graphe d'une fonction feuille potentielle peut être dessiné, qui est colorée en vert. Cette partie du plan est obtenue en utilisant les contraintes de la proposition 2.1.1.

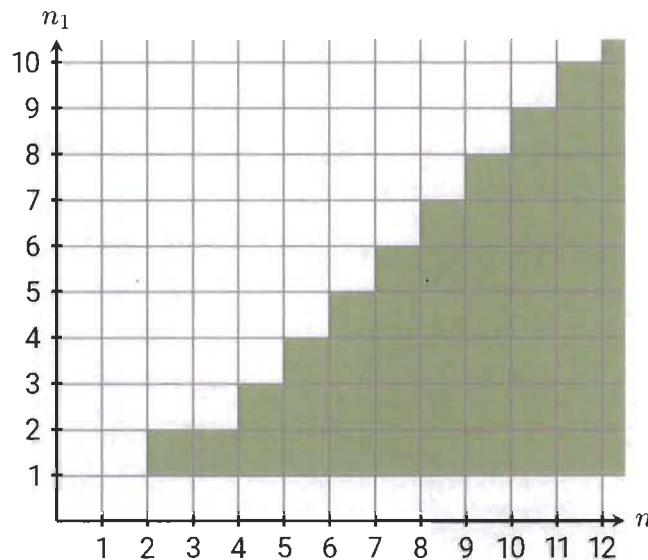


FIGURE 2.4 – Espace des fonctions feuille potentielles

2.2 Méthode de preuve

Le but principal du présent chapitre est de caractériser les sous-arbres induits pleinement feuillus des polyformes et des polycubes. Pour ce faire, on étudie les graphes G_{\square} , G_{\triangle} , G_{\circ} et G_{\ominus} définis à la section 1.1. Les sous-arbres induits de ces graphes sont respectivement des polyominos arbre, des polyhexes arbre, des polytriangles arbre et des polycubes arbre.

Dans le cas où le graphe G est infini, on cherche une expression pour la fonction feuille de G donnant le nombre maximal de feuilles d'un sous-arbre induit de taille n pour tout entier positif $n \geq 2$. Il est impossible de simplement parcourir tous les sous-arbres induits, puisqu'il en existe une infinité.

En particulier dans le cas des quatre graphes précédents, on trouve les fonctions feuille désirées en trois étapes :

1. trouver une fonction feuille potentielle ℓ_G qui est conjecturée égale à la fonction feuille L_G de G ;

2. construire un ensemble de sous-arbres induits de taille n ayant $\ell_G(n)$ feuilles pour toutes les valeurs de $n \geq 2$ pour montrer que ℓ_G est une borne inférieure de L_G ;
3. montrer que ℓ_G est une borne supérieure de L_G par contradiction avec la méthode du contre-exemple minimal.

La première étape de la démonstration se fait en explorant les sous-arbres induits de G pour de petite taille et en comptant leurs feuilles. En généralisant la structure des sous-arbres induits construits, on définit une fonction ℓ_G qui donne le nombre maximal de feuilles conjecturé, c'est-à-dire que l'on conjecture $\ell_G(n) = L_G(n)$, pour tout entier positif $n \geq 2$. Dans toutes les grilles étudiées, les fonctions feuille potentielles proposées sont définies par des récurrences linéaires.

Lorsque l'on définit la fonction feuille potentielle ℓ_G , il est possible de conjecturer des valeurs fausses. Comme on n'étudie que des sous-arbres induits de petites tailles et que l'on généralise pour toutes les tailles, il est possible qu'une structure plus grande nous échappe et ait plus de feuilles que notre conjecture. Un exemple d'une telle erreur est survenu lors de l'étude des polycubes. Les deux étapes suivantes permettent de détecter et de corriger d'éventuelles erreurs.

La deuxième étape se fait à l'aide des sous-arbres induits trouvés à la première étape. On cherche une construction simplifiée d'un ensemble particulier de sous-arbres induits. L'ensemble construit doit avoir un sous-arbre induit à n sommets ayant $\ell_G(n)$ feuilles pour chaque entier positif $n \geq 2$. L'existence de l'ensemble construit montre que le nombre maximal de feuilles dans G est au moins $\ell_G(n)$,

$$\ell_G(n) \leq L_G(n), \forall n \geq 2.$$

La troisième étape est la plus complexe. Elle vise à montrer qu'il n'existe pas de sous-arbre induit de taille n ayant plus de $\ell_G(n)$ feuilles, et donc que $\ell_G \geq L_G$. On montre cette étape par contradiction avec la méthode du contre-exemple minimal.

La preuve par contre-exemple minimal est logiquement similaire à la preuve par induction. Elle permet de montrer qu'une propriété est vraie pour tous les nombres naturels à partir

d'une certaine valeur. Dans ce cas-ci, la propriété que l'on veut montrer est

$$\ell_G(n) \geq L_G(n), \quad \forall n \geq 2.$$

L'avantage principal de la preuve par contre-exemple minimal est qu'elle permet de faire des sauts de taille variable. Alors que dans la preuve par induction on utilise le cas n pour montrer le cas $n + 1$, dans la preuve par contre-exemple minimal, on peut contredire le cas n par le cas $n - i$ pour n'importe quelle valeur de $0 < i < n$. Cette propriété est utile dans l'étude des sous-arbres induits pleinement feuillus, puisque leur structure est souvent répétitive.

Définition 2.2.1 – Fonction d'écart

Soit ℓ , une fonction feuille potentielle. La fonction d'écart de ℓ , notée $\Delta_\ell : \mathbb{N} \rightarrow \mathbb{N}$ est définie par

$$\Delta_\ell(k) = \liminf_{n \geq 2} (\ell(n + k) - \ell(n)).$$

On s'intéresse à Δ_ℓ en particulier lorsque ℓ est définie par une récurrence linéaire simple ou lorsque le graphe est fini. Dans ce cas, Δ_ℓ est bien définie pour toutes les valeurs de k . Dans certains cas, on peut remplacer la limite inférieure par un minimum, ce qui simplifie la définition.

La fonction d'écart permet de caractériser la variation minimale dans la fonction ℓ et d'obtenir le lemme suivant. C'est ce lemme qui est utilisé pour compléter la preuve par contre-exemple minimal.

Lemme 2.2.1

Soit G , un graphe quelconque et ℓ , une fonction feuille potentielle. Soit T et T' , deux sous-arbres induits de G . Si les inégalités suivantes sont respectées,

$$n(T) > n(T'), \quad (2.4)$$

$$n_1(T) > \ell(n(T)), \quad (2.5)$$

$$n_1(T) - n_1(T') \leq \Delta_\ell(n(T) - n(T')), \quad (2.6)$$

alors

$$n_1(T') > \ell(n(T')).$$

Démonstration. L'inégalité se montre directement,

$$\begin{aligned} n_1(T') &\geq n_1(T) - \Delta_\ell(n(T) - n(T')) && \text{par 2.6,} \\ &> \ell(n(T)) - \Delta_\ell(n(T) - n(T')) && \text{par 2.5,} \\ &\geq \ell(n(T')) && \text{par la définition 2.2.1.} \end{aligned}$$

□

On utilise ce lemme en particulier dans le cas où l'on a un sous-arbre induit T d'un graphe G ayant plus de feuilles que $\ell_G(n(T))$ et qui est de taille minimale satisfaisant cette propriété. On construit alors un arbre T' satisfaisant les hypothèses du lemme 2.2.1. Ce lemme nous assure que T' a plus de feuilles que $\ell_G(n(T'))$, contredisant la minimalité de T .

2.3 Sous-arbres induits enracinés figés

Afin d'étudier plus en profondeur les sous-arbres induits d'un graphe G , on les factorise en objets plus simples. En étudiant les parties des sous-arbres induits, on déduit des propriétés des objets en entier.

Définition 2.3.1 – Sous-arbre induit enraciné figé

Un sous-arbre induit enraciné figé d'un graphe G est un triplet $R = (T, x, \{x, y\})$ tel que

1. T est un sous-arbre induit de G ;
2. x est un sommet de T appelé la racine de R ;
3. $\{x, y\}$ est une arête de T appelée la direction de R .

Une arête $\{x, z\}$ est une direction libre de R si z est une feuille dans T . Un sous-arbre induit enraciné figé est final si $\{x, y\}$ n'est pas une direction libre de R .

Les sous-arbres induits enracinés figés sont les parties élémentaires des sous-arbres induits. C'est-à-dire qu'un sous-arbre induit peut être construit à l'aide de sous-arbres induits enracinés figés. On utilise le terme *figé* pour signifier qu'en plus du sommet sur lequel le sous-arbre induit est enraciné, une direction est choisie, qui apporte une spécification supplémentaire à la structure.

Définition 2.3.2 – Profondeur

La profondeur d'un sous-arbre induit enraciné figé $R = (T, x, \{x, y\})$ est la longueur du plus long chemin de la racine de R à une feuille de T . Un sous-arbre induit enraciné figé est atomique s'il est de profondeur 1.

Dans les figures qui suivent, un sous-arbre induit enraciné figé $R = (T, x, \{x, y\})$ est illustré en dessinant l'arbre T , en dessinant une cerce sur le sommet x et en traçant l'arête $\{x, y\}$ avec un trait plus gras et d'une couleur différente. Un sous-arbre induit enraciné figé de profondeur 3 est illustré à la figure 2.5 (a) et un sous-arbre induit enraciné figé atomique est illustré dans la figure 2.5 (b). Les graphes G dans lesquels ces graphes sont induits ne sont pas illustrés pour alléger la figure.

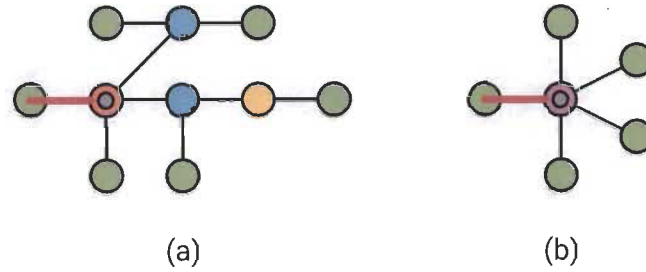


FIGURE 2.5 – Exemples de sous-arbre induit enraciné figé

Par abus, on désigne parfois un sous-arbre induit enraciné figé comme un arbre. Par exemple, lorsque l'on écrit $n(R)$ pour un sous-arbre induit enraciné figé R , on veut dire $n(T)$.

On définit une opération permettant de fusionner deux de ces objets en un autre sous-arbre induit enraciné figé.

Définition 2.3.3 – Greffe

Soit $R = (T, x, \{x, y\})$ et $R' = (T', x', \{x', y'\})$, deux sous-arbres induits enracinés figés d'un graphe G et $\{x, z\}$ une direction libre de R . Soit τ , un automorphisme de G déplaçant le sommet x' sur le sommet z et le sommet y' sur le sommet x . Notons S et S' , les ensembles de sommets de T et T' respectivement, tels que $G[S] = T$ et $G[S'] = T'$.

La greffe de R' sur R , notée $R \triangleleft R'$, lorsqu'elle est bien définie, est le sous-arbre induit enraciné figé défini par

$$R \triangleleft R' = (G[S \cup \tau(S')], x, \{x, y\}).$$

En général, la greffe n'est pas commutative, c'est-à-dire que $R \triangleleft R'$ ne donne pas toujours le même résultat que $R' \triangleleft R$. Le résultat de la greffe de deux sous-arbres induits enracinés figés n'est pas toujours un sous-arbre induit enraciné figé. En particulier, l'ajout de sommets peut créer un cycle. Une greffe $R \triangleleft R' = (T, x, \{x, y\})$ est dite

1. Bien définie si le sous-graphe induit $G[S \cup \tau(S')]$ est un arbre ;
2. Finale si $R \triangleleft R'$ est un sous-arbre induit enraciné figé final ;
3. invalide sinon, dans le cas où l'on a créé un cycle par exemple, s'il n'existe pas d'automorphisme τ ayant les propriétés désirées ou si R n'a pas de direction libre appropriée.

Un exemple de greffe bien définie non finale est donnée dans la figure 2.6.

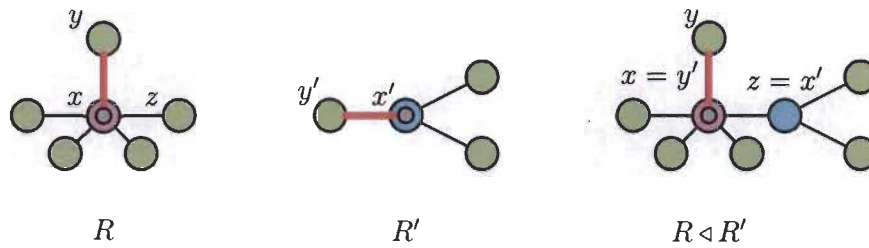


FIGURE 2.6 – Exemple de greffe de sous-arbres induits enracinés figés

Le plus petit sous-arbre enraciné figé possible est le sous-arbre induit enraciné figé identité formé de deux sommets adjacents dans un graphe x et y , $e = (G[\{x, y\}], x, \{x, y\})$. La greffe de e sur un sous-arbre induit enraciné figé R redonne exactement R , c'est l'élément neutre de l'opération de greffe. Cette procédure est illustrée dans la figure 2.7.

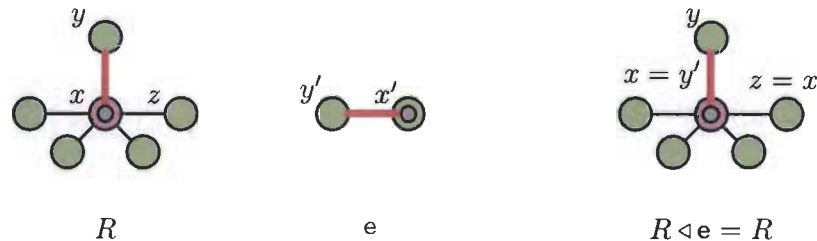


FIGURE 2.7 – Greffe du sous-arbre induit enraciné figé identité

Comme le résultat d'une greffe est un sous-arbre induit enraciné figé, il est possible d'appliquer la greffe sur le résultat d'une greffe. Par exemple, considérons cinq sous-arbres induits

enracinés figés R_1, R_2, R_3, R_4, R_5 la greffe suivante est possiblement bien définie

$$R_1 \triangleleft R_2 \triangleleft R_3 \triangleleft R_4 \triangleleft R_5.$$

L'ordre d'opération dans une greffe successive est avec une priorité à gauche, c'est-à-dire

$$R_1 \triangleleft R_2 \triangleleft R_3 \triangleleft R_4 \triangleleft R_5 = (((R_1 \triangleleft R_2) \triangleleft R_3) \triangleleft R_4) \triangleleft R_5.$$

Certaines propriétés sur le nombre de sommets dans le sous-arbre enraciné figé résultant d'une greffe sont utiles pour la suite, et sont obtenues directement de la définition.

Proposition 2.3.1

Soit R et R' deux sous-arbres induits enracinés figés tels que $R \triangleleft R'$ est bien définie.

On a les propriétés suivantes sur les degrés et la taille du résultat d'une greffe,

$$n(R \triangleleft R') = n(R) + n(R') - 2$$

$$n_1(R \triangleleft R') = n_1(R) + n_1(R') - 2,$$

$$n_i(R \triangleleft R') = n_i(R) + n_i(R'), \forall i > 1.$$

Démonstration. Ces propriétés se déduisent directement de la définition de greffe. Considérons deux sous-arbres induits $R = (T, x, \{x, y\})$ et $R' = (T', x', \{x', y'\})$ d'un graphe G tel que $\{x, z\}$ est une direction libre de R . Soit τ un automorphisme de G déplaçant x' sur z et y' sur x . Comme la greffe est bien définie, $R \triangleleft R'$ est un arbre. On a donc que les seuls sommets superposés sont x' sur z et y' sur x . Les feuilles z et y' sont donc remplacées par les sommets x' et x .

Deux feuilles sont disparues. Tous les autres sommets sont inchangés par la greffe, puisque ce n'est qu'une union de sommets. \square

Par exemple, dans la figure 2.6, les statistiques sur les sous-arbres induits enracinés figés

R , R' et $R \triangleleft R'$ sont

$$\begin{array}{llll} n_1(R) = 5, & n_3(R) = 0, & n_5(R) = 1, & n(R) = 6 \\ n_1(R') = 3, & n_3(R') = 1, & n_5(R') = 0, & n(R') = 4 \\ n_1(R \triangleleft R') = 6, & n_3(R \triangleleft R') = 1, & n_5(R \triangleleft R') = 1, & n(R \triangleleft R') = 8. \end{array}$$

La profondeur du résultat d'une greffe est prévisible. Si l'on greffe un sous-arbre induit enraciné figé R' de profondeur p' sur un sous-arbre induit enraciné figé R de profondeur p , la profondeur est donnée par $\max\{p, p' + 1\}$. En effet, si le plus long chemin était déjà présent dans R , la profondeur reste p , sinon le plus long chemin passe dans R' et est allongé de 1, puisque la racine est maintenant dans R .

Définition 2.3.4 – Branche

Soit T , un sous-arbre induit d'un graphe G . Un sous-arbre induit R est une *branche* de T s'il existe un sous-arbre induit R^c tel que

$$R^c \triangleleft R = (T, x, \{x, y\}).$$

Le sous-arbre induit R^c est appelé la *co-branche* de R dans T .

En particulier, tout sous-arbre induit T ayant au moins deux sommets contient le sous-arbre induit enraciné figé identité.

Il est possible de factoriser un sous-arbre induit T dans un graphe G en une paire branche, co-branche. Soit x, y deux sommets adjacents dans T . Notons V_x et V_y , les ensembles de sommets de T qui sont les deux ensembles connexes de sommets contenant respectivement x et y obtenus en retirant l'arête $\{x, y\}$ de l'ensemble de sommets de T .

Le sous-arbre induit enraciné figé $R = (T[V_x \cup \{y\}], x, \{x, y\})$ est une branche de T et $R^c = (T[V_y \cup \{x\}], y, \{x, y\})$ est la co-branche de R dans T . La construction d'une paire branche, co-branche dans un arbre est illustrée à la figure 2.8.

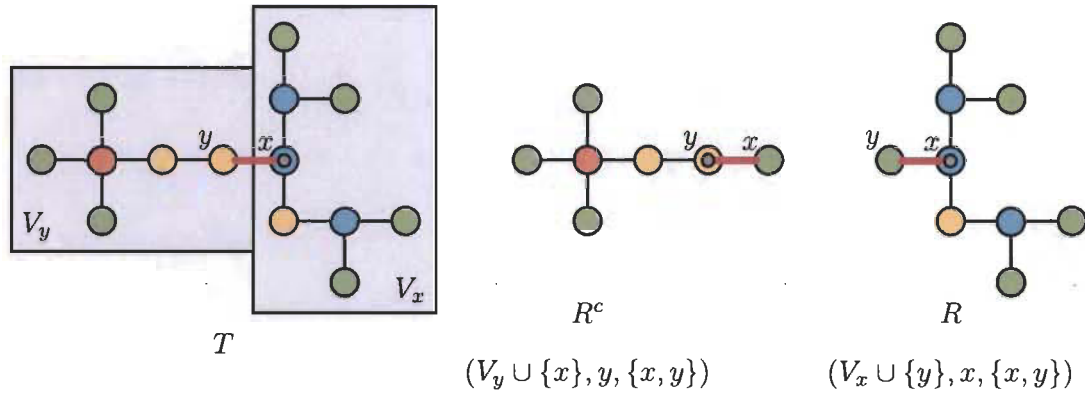


FIGURE 2.8 – Exemple de la factorisation d'un arbre en une paire branche, co-branche

Définition 2.3.5 – Substitution

Soit T , un sous-arbre induit d'un graphe G et soit R et R^c , une paire branche, co-branche de T . Soit R' , un sous-arbre induit enraciné figé du graphe G .

La substitution de R par R' dans T , notée $T' = T \triangleleft_R R'$, est l'arbre qui est le premier élément du triplet de la greffe de R' sur R^c ,

$$(T', x, \{x, y\}) = R^c \triangleleft R'.$$

La substitution est une opération composée. On fractionne d'abord un arbre en une paire branche, co-branche puis l'on greffe un sous-arbre induit enraciné figé sur la co-branche. On peut voir la substitution comme le remplacement d'une branche R par un sous-arbre induit enraciné figé R' dans un arbre T . Un exemple de substitution est donné dans la figure 2.9.

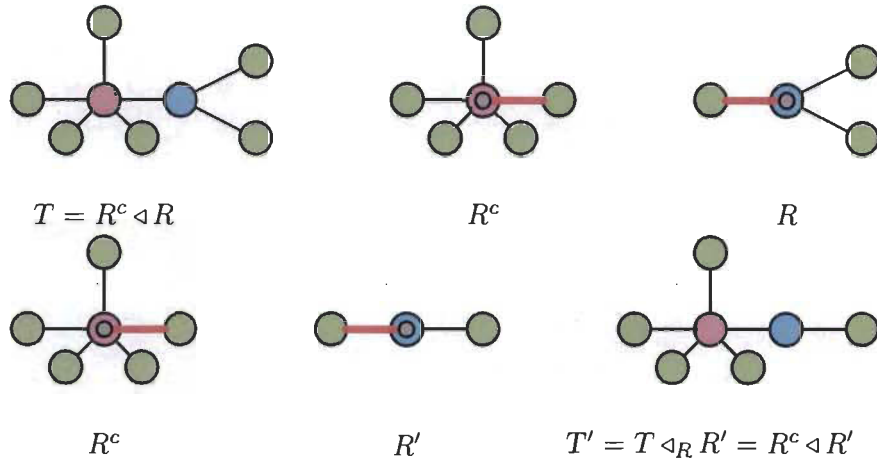


FIGURE 2.9 – Substitution d'une branche dans un sous-arbre induit

On obtient directement les résultats suivants sur la substitution en utilisant la proposition 2.3.1 sur le nombre de sommets de degré donné dans le résultat d'une greffe.

Corollaire 2.3.1

Soit T un sous-arbre induit d'un graphe G et R une branche de T . Soit R' , un sous-arbre induit enraciné figé de G tel que $T' = T \triangleleft_R R'$ est bien définie. On a les propriétés suivantes sur les degrés de T' .

$$n(T) - n(T') = n(R) - n(R'),$$

$$n_i(T) - n_i(T') = n_i(R) - n_i(R'), \forall i \geq 1.$$

Démonstration. Comme $T = R^c \triangleleft R$, on a

$$n(T) = n(R^c) + n(R) - 2.$$

De plus, puisque $T' = R^c \triangleleft R'$,

$$n(T') = n(R^c) + n(R') - 2.$$

En soustrayant ces deux équations, on obtient

$$\begin{aligned} n(T) - n(T') &= n(R^c) + n(R) - 2 - (n(R^c) + n(R') - 2), \\ &= n(R) - n(R'). \end{aligned}$$

□

Comme la substitution est définie par une greffe, tous les qualificatifs de la greffe sont utilisés pour la substitution. On a ainsi une substitution bien définie et une substitution finale.

Définition 2.3.6 – Sous-arbre induit enraciné figé substituable

Soit R et R' , deux sous-arbres induits enracinés figés dans un graphe G . Si pour tout sous-arbre induit T dans G contenant R comme branche, la substitution de R par R' est bien définie, on dit que R est substituable par R' .

Définition 2.3.7 – Parent et enfant

Soit $R = (T, x, \{x, y\})$, un sous-arbre induit enraciné figé d'un graphe G . Le sous-arbre induit enraciné figé R' est un enfant de R si R' est une branche de R qui ne contient pas le sommet y . Le sous-arbre induit enraciné R est un parent de R .

Si un enfant R' de R contient la racine x de R , alors R' est un enfant direct de R et R est un parent direct de R' .

Un exemple de tous les enfants directs d'un sous-arbre induit enraciné figé est donné dans la figure 2.10.

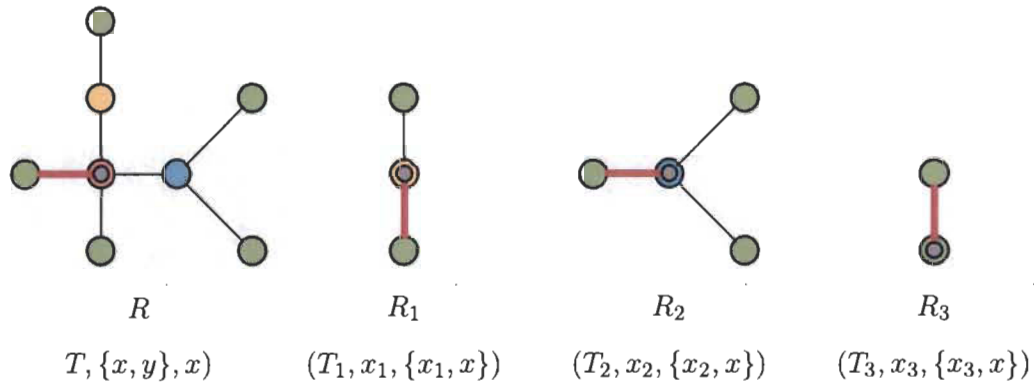


FIGURE 2.10 – Les trois enfants directs d'un sous-arbre induit enraciné figé

Si un sous-arbre induit enraciné figé R est de profondeur p , alors la profondeur d'un enfant R' de R est au plus $p - 1$. De plus, au moins un enfant direct de R est de profondeur $p - 1$. Ces observations sont des conséquences directes de la définition de profondeur, pour avoir un chemin de longueur p , il doit y voir un chemin de longueur $p - 1$ lorsque l'on retire une cellule.

Un sous-arbre induit enraciné figé R de profondeur p dont la racine est de degré d peut toujours être construit en greffant ses enfants directs sur un sous-arbre induit enraciné figé atomique ayant $d + 1$ sommets. On s'intéresse en particulier à l'ensemble R_p des sous-arbres induits enracinés figés de profondeur p d'un graphe G .

Définition 2.3.8

Soit ℓ , une fonction feuille potentielle. Le sous-arbre induit enraciné figé R d'un graphe G est épars par rapport à ℓ s'il existe un sous-arbre induit enraciné figé R' tel que R est substituable par R' , que $n(R') < n(R)$ et

$$n_1(R) - n_1(R') \leq \Delta_\ell(n(R) - n(R')).$$

Un sous-arbre induit enraciné figé qui n'est pas épars par rapport à ℓ est abondant.

En particulier, tout sous-arbre induit enraciné figé R est substituable par e , on trouve une condition nécessaire, mais non suffisante pour que R soit épars. R est épars par rapport à ℓ si

$$n_1(R) - n_1(e) \leq \Delta_\ell(n(R) - n(e)),$$

$$n_1(R) - 2 \leq \Delta_\ell(n(R) - 2).$$

Cette condition permet d'identifier rapidement certains sous-arbres induits enracinés figés épars par rapport à ℓ .

Une condition géométrique permet de déterminer quand un sous-arbre induit enraciné figé est substituable par un autre.

Définition 2.3.9 – Coque

L'extension d'un sous-graphe induit G' d'un graphe G , notée $\text{Ext}(G')$, est le sous-graphe induit par l'ensemble des voisins de tous les sommets de G' .

L'intérieur d'un sous-graphe induit G' d'un graphe G , noté $\text{Int}(G')$, est l'ensemble des sommets dont tous les voisins sont inclus dans G' .

La coque d'un sous-graphe G' est l'intérieur de l'extension de G' .

Pour illustrer le concept de coque, on utilise la grille carrée comme graphe. Considérons le sous-arbre induit P de G_\square illustré à la figure 2.11 (a). Son extension est dessinée dans la figure 2.11 (b) et l'intérieur de son extension, qui est sa coque, est dans la figure 2.11 (c).

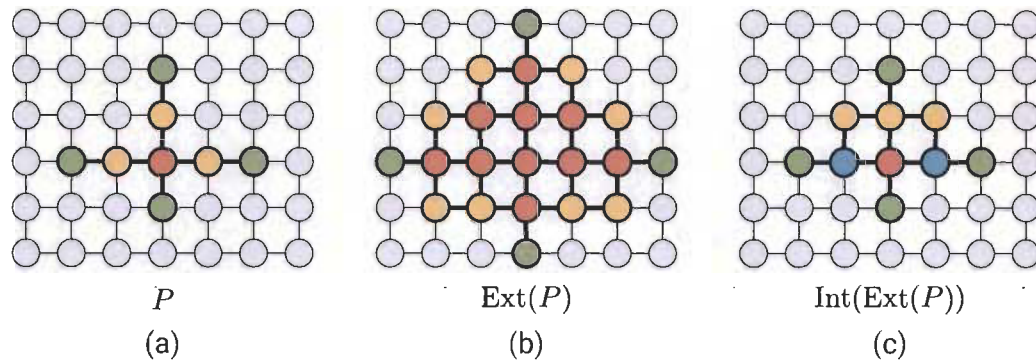


FIGURE 2.11 – La coque d'un sous-graphe induit de la grille carrée

Proposition 2.3.2

Soit $R = (T, x, \{x, y\})$ et $R' = (T', x', \{x', y'\})$, deux sous-arbres induits enracinés figés d'un graphe G et soit τ , un automorphisme de G déplaçant x sur x' et y sur y' . Si tous les sommets la coque de $\tau(T' - \{y'\})$ sont inclus dans la coque de $T - \{y\}$, alors R est substituable par R' .

Démonstration. Un sous-arbre induit enraciné figé R est substituable par un sous-arbre induit enraciné figé R' si la substitution de R par R' dans un sous-arbre induit T est toujours bien définie, peu importe le sous-arbre induit T , c'est-à-dire que la substitution ne crée jamais de cycle.

En particulier, un cycle est créé lors d'une substitution si un sommet du sous-arbre enraciné figé R' qui n'est pas sa racine est adjacent à un sommet de R^c .

Supposons que la substitution ne donne pas un arbre et que $T' - \{y'\}$ soit incluse dans la coque de $T - \{y\}$. Il existe alors un sommet c de R' qui est adjacent à un sommet c' de R^c . Comme c est dans la coque de $T - \{y\}$, le sommet c' voisin de c est dans l'extension de $T - \{y\}$. Le sommet c était donc présent dans R et on avait déjà un cycle, ce qui est une contradiction. \square

Lemme 2.3.1

Soit G un graphe et ℓ , une fonction feuille potentielle. Si un sous-arbre induit T de G contient un sous-arbre induit enraciné figé R comme branche qui est épars par rapport à ℓ par le sous-arbre induit enraciné figé R' de G et si T a plus de $\ell(n(T))$ feuilles, alors $T' = T \triangleleft_R R'$ a plus de $\ell(n(T'))$ feuilles.

Démonstration. Comme R est épars par rapport à ℓ par R' , on a

$$n_1(R) - n_1(R') \leq \Delta_\ell(n(R) - n(R')).$$

Sachant que

$$\begin{aligned} n(T) &> n(T'), \\ n_1(T) &> \ell(n(T)), \end{aligned}$$

on applique le corollaire 2.3.1 pour obtenir

$$\begin{aligned} n_1(T) - n_1(T') &= n_1(R) - n_1(R') \\ &\leq \Delta_\ell(n(R) - n(R')) \\ &= \Delta_\ell(n(T) - n(T')). \end{aligned}$$

Par le lemme 2.2.1, on a $n_1(T') > \ell(n(T'))$. □

Ceci précise la méthode de démonstration de l'exactitude d'une fonction feuille. On veut montrer que dans un graphe G , il existe une profondeur k telle que tous les sous-arbres induits enracinés figés de G qui sont de profondeur au moins k sont épars. Pour ce faire, on montre que tous les parents d'un sous-arbre induit enraciné figé abondant sont épars à partir d'une certaine profondeur.

Proposition 2.3.3

Soit R un sous-arbre induit enraciné figé abondant. Tous les enfants de R sont abondants.

Démonstration. Considérons un sous-arbre induit enraciné figé R ayant un enfant R_1 qui est épars par le sous-arbre induit enraciné figé R_2 . Considérons $R' = R \triangleleft_{R_1} R_2$. On a donc

$$\begin{aligned} n_1(R) - n_1(R') &= n_1(R_1) - n_1(R_2) \\ &\geq \Delta(n(R_1) - n(R_2)) \\ &= \Delta(n(R) - n(R')). \end{aligned}$$

Donc le sous-arbre induit enraciné figé R est épars par R' . \square

On dit qu'un sous-arbre induit enraciné figé est **potentiellement abondant** si tous ses enfants sont abondants.

Dans un graphe G , notons \mathcal{P}_p et \mathcal{A}_p , les ensembles des sous-arbres induits enracinés figés potentiellement abondants et abondants de profondeur p . On décrit une méthode de construction pour ces ensembles.

Supposons que l'on a construit l'ensemble des sous-arbres induits enracinés figés abondants de G jusqu'à la profondeur $p - 1$, c'est-à-dire \mathcal{A}_{p-1} . L'ensemble des sous-arbres induits enracinés figés potentiellement abondants de profondeur p , \mathcal{P}_p est obtenu en prenant un sous-arbre induit enraciné figé atomique dont la racine est de degré d et en y greffant un sous-arbre induit enraciné figé abondant de profondeur $p - 1$ et $d - 1$ sous-arbres induits enracinés figés de profondeur inférieure à p sur chacune de ses feuilles.

On obtient ensuite les sous-arbres induits enracinés figés abondants de profondeur p \mathcal{A}_p en choisissant les éléments de \mathcal{P}_p qui ne sont ni épars, ni finaux, selon la définition 2.3.8.

Afin de simplifier l'écriture d'un sous-arbre induit enraciné figé R , on définit une notation décrivant sa structure. On définit la fonction w sur R appelée l'**arborescence des degrés** de R par $w(R) = \mathbf{d}$ si R est un sous-arbre induit enraciné figé atomique dont la racine est de degré d et par

$$w(R) = \mathbf{d}(w(R_1), w(R_2), \dots, w(R_k)).$$

si la racine de R est de degré d et si R est obtenu de la greffe des sous-arbres induits enracinés figés R_1, R_2, \dots, R_k sur une racine de degré d . On utilise une police à chasse fixe

pour différencier le sous-arbre induit enraciné figé 3 du nombre 3. Un exemple de cette notation est donné dans la figure 2.12.

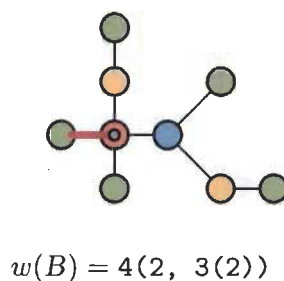


FIGURE 2.12 – Exemple de l'arborescence des degrés d'un sous-arbre induit enraciné figé

En effet, deux sous-arbre induits enracinés figés peuvent avoir la même arborescence des degrés. Par exemple, dans la grille carrée, il existe deux sous-arbres induits enracinés figés atomiques dont le degré de la racine est 3. La notation des deux objets est la même, $w(R) = 3$, mais les polyominos arbre enracinés figés sont différents en tant que sous-arbres induits d'un graphe G . Ces deux sous-arbres induits enracinés figés sont illustrés dans la figure 2.13.

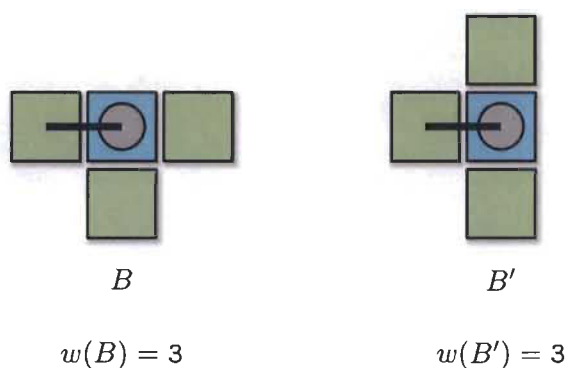


FIGURE 2.13 – Deux polyominos arbre enracinés figés distincts ayant la même notation

On utilise donc uniquement l'arborescence de degrés lorsque l'on s'intéresse seulement au

nombre de feuilles et à la taille des sous-arbres induits enracinés figés. En effet, il est plutôt facile de trouver le nombre de feuilles et la taille d'un sous-arbre induit enraciné figé en utilisant seulement son arborescence des degrés. Considérons un sous-arbre induit enraciné figé R dont l'arborescence des degrés $w(R)$ contient les nombres d_i , pour $d_i > 1$. On a alors les égalités suivantes,

$$\begin{aligned} n(w(R)) &= n(R) = \sum_i (d_i - 1) + 2, \\ n_1(w(R)) &= n_1(R) = \sum_i (d_i - 2) + 2. \end{aligned}$$

2.4 Saturation

Certains sous-arbres induits pleinement feuillus d'un graphe G ont une structure plus restreinte que d'autres. En effet, lorsque l'augmentation du nombre de feuilles n'est pas régulière, la fonction feuille de G a parfois une longue période d'augmentation suivie d'une période de stagnation. Les sous-arbres induits qui maximisent le ratio $\frac{n_1}{n}$ du nombre de feuilles sur le nombre total de cellules sont ceux qui nous intéressent.

Pour la suite, on considère les fonctions feuille potentielles qui sont définies par des cas de base et une partie récursive. C'est-à-dire une fonction ayant un certain nombre r de cas de base pour $2 \leq n \leq r + 1$ et une règle de récurrence qui donne une fonction périodique de période q . En général, on peut écrire une telle fonction ℓ sous la forme suivante,

$$\ell(n) = \begin{cases} \ell(n - q) + a & \text{si } n > r + 1, \\ b_n & \text{si } 2 \leq n \leq r + 1. \end{cases}$$

On considère que a est aussi un nombre naturel, on appelle a/q la pente de la fonction ℓ . Les nombres b_n sont des nombres entiers qui donnent les valeurs des cas de base de la fonction ℓ .

On dit qu'une fonction qui est définie de cette façon suit une récurrence linéaire simple, c'est-à-dire une récurrence linéaire non homogène à coefficients constants ayant un seul

terme. On utilise cette définition parce que les fonctions feuille des grilles étudiées dans les chapitres 3, 4 et 5 sont définies de cette façon.

Proposition 2.4.1

Soit G , un graphe dont la fonction feuille L_G est définie par une récurrence linéaire simple. Il existe deux uniques fonctions $\overline{L_G}$ et $\underline{L_G}$ et un entier k tels que pour tout entier positif $n > k$,

$$\underline{L_G}(n) \leq L_G(n) \leq \overline{L_G}(n), \forall n > k$$

et tels qu'il existe une infinité d'entiers positifs n, n' pour lesquels

$$\underline{L_G}(n') = L_G(n')$$

$$\overline{L_G}(n) = L_G(n).$$

Démonstration. Comme L_G est définie par une récurrence linéaire simple à coefficients constants, disons de période r et de pente $\frac{a}{r}$, on peut réécrire L_G comme suit :

$$L_G(n) = \begin{cases} \frac{a+b_0}{r} \cdot n & \text{si } n \equiv 0 \pmod{k}, \\ \frac{a+b_1}{r} \cdot n & \text{si } n \equiv 1 \pmod{k}, \\ \vdots \\ \frac{a+b_{k-1}}{r} \cdot n & \text{si } n \equiv k-1 \pmod{k}. \end{cases}$$

Les fonctions $\underline{L_G}$ et $\overline{L_G}$ sont donc définies par

$$\underline{L_G}(n) = \frac{a + \min_{i \geq 2} \{b_i\}}{r} n,$$

$$\overline{L_G}(n) = \frac{a + \max_{i \geq 2} \{b_i\}}{r} n.$$

Notons i_{\min} et i_{\max} , les indices tels que

$$b_{i_{\min}} = \min_{i \geq 2} \{b_i\},$$

$$b_{i_{\max}} = \max_{i \geq 2} \{b_i\}.$$

On a alors

$$\underline{L}_G(n) = L_G(n) \iff n \equiv i_{\min} \pmod{q},$$

$$\overline{L}_G(n) = L_G(n) \iff n \equiv i_{\max} \pmod{q}.$$

C'est-à-dire,

$$\underline{L}_G(qk + i_{\min}) = L_G(qk + i_{\min}),$$

$$\overline{L}_G(qk + i_{\max}) = L_G(qk + i_{\max}).$$

□

Définition 2.4.1 – Sous-arbre induit saturé

Soit G , un graphe quelconque dont la fonction feuille L_G est définie par une récurrence linéaire simple. Un sous-arbre induit $T \in \mathcal{T}_G$ est dit **saturé** s'il a au moins autant de feuilles que $\overline{L}_G(n(T))$.

On étudie les sous-arbres induits saturés dans cette section. On peut parfois les énumérer ou parfois caractériser leur structure, puisqu'elle est plus restreinte. Lorsqu'un sous-arbre induit pleinement feuillu n'ayant pas cette caractéristique est dit **non saturé**.

Comme avec les sous-arbres induits pleinement feuillus, il est plus utile d'étudier les sous-arbres induits enracinés figés qui peuvent être une branche dans un sous-arbre induit saturé. L'utilisation de ces objets permet d'utiliser une forme d'induction sur les sous-arbres induits saturés.

Définition 2.4.2 – Sous-arbre induit enraciné figé saturé

Un sous-arbre induit enraciné figé R d'un graphe G est dit **saturé** s'il existe un sous-arbre induit T saturé de G contenant R comme branche.

Afin de caractériser les arbres saturés, on étudie les sous-arbres induits saturés, qui donnent toute l'information nécessaire sur les sous-arbres induits saturés. En particulier, comme on connaît l'aire des sous-arbres saturés et la fonction feuille du graphe, on peut utiliser le nombre de feuilles pour caractériser les sous-arbres induits enracinés figés saturés.

Définition 2.4.3 – Écart saturé

Soit L_G , la fonction feuille d'un graphe G dont les arbres saturés sont d'aire $qk + r$ avec q et r entiers constants. C'est-à-dire que $L_G(qk + r) = \overline{L_G}(qk + r)$ pour tout entier positif k .

La fonction d'écart saturé de G , notée $\sigma_{L_G}(m)$ est définie pour $m > 0$ entier par

$$\sigma_{L_G}(m) = L_G(qk + r) - L_G(qk + r - m),$$

pour k tel que $qk + r > m + 2$.

La fonction d'écart saturé donne exactement le nombre de feuilles que l'on perd en retirant une à une les feuilles d'un sous-arbre induit saturé en s'assurant de conserver un arbre pleinement feuillu à chaque étape. On utilise cette fonction pour caractériser les sous-arbres induits enracinés figés. Le fait que l'on connaisse les valeurs de L_G nous permet d'obtenir aisément les valeurs de σ_{L_G} , et donc de trouver les sous-arbres induits saturés.

Proposition 2.4.2

Soit R , un sous-arbre induit enraciné figé d'un graphe G . S'il existe un sous-arbre induit enraciné figé R' telle que R est substituable par R' et

$$\sigma_{L_G}(n(R) - n(R')) > n_1(R) - n_1(R'),$$

alors R n'est pas un sous-arbre induit enraciné figé saturé.

Démonstration. Supposons que R soit un sous-arbre induit enraciné figé saturé de G . Il existe donc un sous-arbre induit saturé T dans G contenant R comme branche. Soit L_G , la fonction feuille de G , dont les sous-arbres induits saturés sont d'aire $qk + r$ pour tout entier positif k .

Comme T est saturé, on a

$$\begin{aligned} n(T) &= qk + r, \\ n_1(T) &= L_G(qk + r), \end{aligned}$$

pour un certain entier positif k . Posons $T' = T \triangleleft_R R'$ pour un sous-arbre induit enraciné figé R' tel que R est substituable par R' . On a donc

$$\begin{aligned} n_1(T') &= n_1(T) - n_1(R) + n_1(R') && \text{du corollaire 2.3.1} \\ &> L_G(qk + r) - \sigma_{L_G}(n(R) - n(R')) && T \text{ est saturé} \\ &= L_G(qk + r) - L_G(qk + r) + L_G(qk + r - n(R) + n(R')) && \text{de la définition 2.4.3} \\ &= L_G(n(T) - n(R) + n(R')) && \text{taille de } T \\ &= L_G(n(T')) && \text{du corollaire 2.3.1.} \end{aligned}$$

On obtient une contradiction par la définition de fonction feuille 2.1.1, puisque $n_1(T') > L_G(n(T'))$, donc R n'est pas saturé. \square

Cette proposition permet une caractérisation faible des sous-arbres induits enracinés figés saturés.

Définition 2.4.4 – Sous-arbre induit enraciné figé potentiellement saturé

Soit R , un sous-arbre induit enraciné figé dans un graphe G dont la fonction feuille est L_G . Si pour tout sous-arbre induit enraciné figé R' telle que R est substituable par R' on a

$$\sigma_{L_G}(n(R) - n(R')) \leq n_1(R) - n_1(R'),$$

alors R est potentiellement saturé.

Un sous-arbre induit enraciné figé est potentiellement saturé s'il a le nombre feuilles approprié pour sa taille. C'est une condition nécessaires mais pas suffisante pour être saturé.

Pour caractériser les sous-arbres induits saturés, on cherche à énumérer tous les sous-arbres induits enracinés figés potentiellement saturés, et par le fait même les sous-arbres induits

saturés, d'une grille. Par contre, il existe une infinité de sous-arbres induits enracinés figés saturés dans un graphe infini puisqu'il existe des sous-arbres induits saturés arbitrairement grands.

Il est souvent possible de caractériser tous les sous-arbres induits enracinés figés saturés d'un graphe puisque, dans la plupart des cas, les sous-arbres induits saturés ont une certaine répétition dans leur structure. On peut alors trouver des structures qui permettent la construction de sous-arbres induits saturés arbitrairement grands.

Définition 2.4.5 – Sous-arbre induit enraciné figé récursivement saturé

Soit R , un sous-arbre induit enraciné figé dans un graphe G dont la fonction feuille est L_G et dont les sous-arbres induits saturés sont d'aire $n = qk + r$ pour tout entier positif k .

S'il existe un sous-arbre induit potentiellement saturé R' tel que R est substituable par R' , que R' est un enfant de R , que $n(R) - n(R')$ est un multiple de q et que $n_1(R) - n_1(R') = \sigma_{L_G}(n(R) - n(R'))$, alors R est récursivement saturé par R' .

En particulier, si un sous-arbre induit enraciné figé saturé R dans un arbre T est récursivement saturé par un sous-arbre induit enraciné figé potentiellement saturé R' , alors $T' = T \triangleleft_R R'$ est un sous-arbre induit saturé. Si dans un graphe G il existe une profondeur p à laquelle tous les sous-arbres induits enracinés figés saturés de profondeur p sont récursivement saturés, alors on peut trouver la structure des sous-arbres induits saturés.

Considérons un sous-arbre induit saturé T . Toutes ses branches sont des sous-arbres induits enracinés figés saturés. Considérons un sous-arbre induit enraciné figé R qui est une branche de T récursivement saturé par un sous-arbre induit enraciné figé R' . En substituant R par R' dans T , on obtient un arbre T' saturé. La branche R' dans T' est contenue dans une branche R . On peut donc répéter la procédure jusqu'à ce qu'il ne reste plus de cellules. C'est-à-dire que les branches du sous-arbre induit saturé T sont des copies du sous-arbre induit enraciné figé R .

Comme avec les sous-arbres induits enracinés figés abondants, il est assez simple de montrer que tous les enfants d'un sous-arbre induit enraciné figé potentiellement saturé sont potentiellement saturés, en utilisant simplement la définition et la même méthode de preuve que pour la proposition 2.3.3. On peut donc construire l'ensemble de tous les sous-arbres induits enracinés figés saturés d'une certaine profondeur en utilisant la même procédure que celle pour construire les sous-arbres induits enracinés figés abondants. C'est-à-dire que l'on construit tous les sous-arbres induits enracinés figés potentiellement saturés de profondeur p en greffant des sous-arbres induits enracinés figés potentiellement saturés de profondeurs inférieurs sur un sous-arbre induit enraciné figé atomique.

Chapitre 3

Polyominos pleinement feuillus

Savoir.

Surprendre.

— UQTR

Dans ce chapitre, on étudie les polyominos arbre pleinement feuillus. En ce sens, on trouve la fonction feuille de la grille carrée à la section 2.1 et on montre son exactitude à la section 2.2. À la section 2.3, on étudie les polyominos arbre saturés, qui sont une famille plus restreinte de polyominos arbre pleinement feuillus.

Les polyominos arbre pleinement feuillus ont été introduits lors d'une tentative d'énumération des polyominos. En séparant les polyominos arbre de taille fixée selon le nombre de feuilles, on en obtient une partition. En énumérant tous les polyominos arbre de taille n ayant k feuilles pour chaque entier positif k , on énumère tous les polyominos arbre de taille n .

Le nombre minimal de feuilles qu'un polyomino arbre peut avoir est 2. C'est la plus petite valeur de k à étudier peu importe la taille n à laquelle on s'intéresse. Le nombre maximal de feuilles d'un polyomino arbre d'aire n est moins évident à trouver. Dans ce chapitre, on s'intéresse au nombre maximal de feuilles que peut avoir un polyomino arbre d'aire donnée.

Certains résultats permettent de compter des familles de polyominos arbre ayant un nombre de feuilles particulier. En général, on ne connaît aucune méthode permettant d'énumérer les polyominos arbre d'aire n ayant k feuilles. La méthode proposée ici pour énumérer tous les polyominos arbre n'est donc pas utilisable jusqu'à maintenant.

Remarquons que pour deux cellules adjacentes dans un polyomino arbre, la somme des degrés de ces deux cellules est au maximum 6. Par exemple, une cellule de degré 4 ne peut pas être adjacente à une cellule de degré 3 ou plus. Si l'on essaie d'ajouter une cellule, on obtient nécessairement un cycle. Dans la figure 3.1 (a), (b), (c) et (d), toutes les paires de cellules adjacentes possibles sont illustrées, selon leur degré. En particulier, les cas (c) et (d) sont maximaux au sens où l'on ne peut ajouter de cellule sur aucune des deux cellules. Dans la figure 3.1 (e) et (f), on illustre les deux paires de cellules créant systématiquement un cycle lorsqu'elles sont adjacentes. Cette contrainte plutôt simple est utile pour éliminer des cas dans une activité d'énumération.

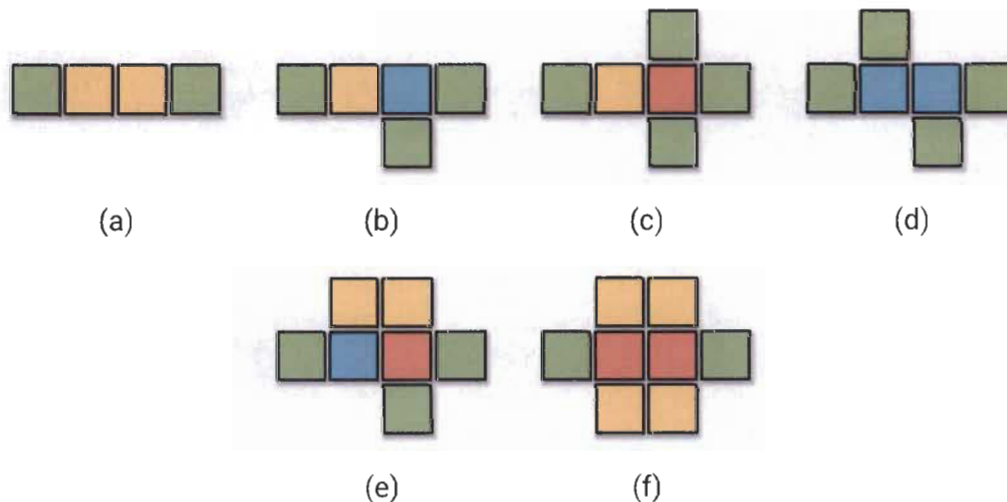


FIGURE 3.1 – Paires de cellules adjacentes pouvant exister dans un polyomino arbre

On note respectivement $\mathcal{P}_{\square}(n)$ et $\mathcal{T}_{\square}(n)$ l'ensemble des polyominos fixés et des polyominos arbre fixés d'aire n . On note par $p_{\square}(n)$ et $t_{\square}(n)$ les cardinalités de ces ensembles.

3.1 Fonction feuille

On s'intéresse au nombre maximal de feuilles dans un polyomino arbre sous l'angle du problème des sous-arbres induits pleinement feuillus du graphe G_{\square} . En modifiant la définition 2.1.1, on obtient la fonction feuille du graphe G_{\square} . Pour simplifier la notation, on pose $L_{\square} = L_{G_{\square}}$ qui est définie pour tout entier positif n par

$$L_{\square}(n) = \max\{n_1(T) : T \in \mathcal{T}_{\square}(n)\}.$$

La fonction feuille L_{\square} permet d'identifier un certain ensemble de polyominos arbre ayant le nombre maximal de feuilles. Un polyomino arbre T est **pleinement feuillu** si

$$n_1(T) = L_{\square}(n(T)).$$

Par exemple, parmi les polyominos arbre d'aire 5, le nombre maximal de feuilles est 4. On a donc

$$L_{\square}(5) = 4.$$

En effet, en étudiant tous les polyominos arbre d'aire 5, à symétrie près, on en trouve sept ayant deux feuilles, trois ayant trois feuilles et un seul ayant quatre feuilles. Ces polyominos sont dessinés à la figure 3.2.

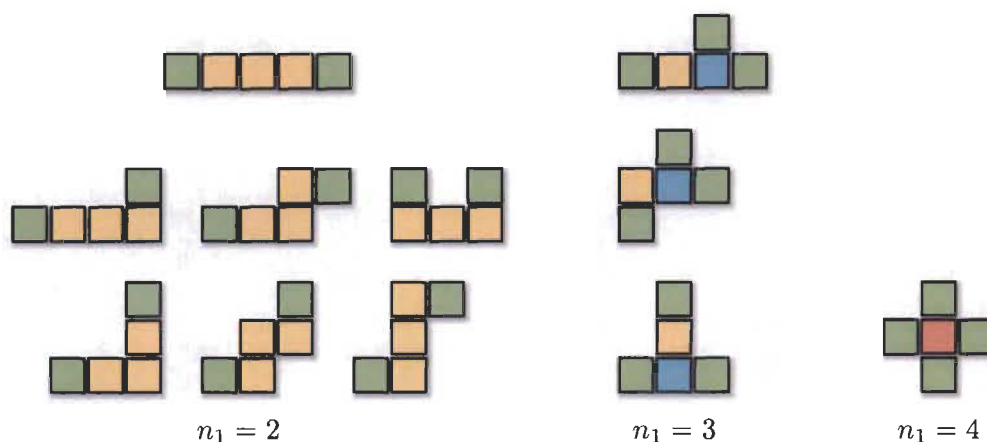


FIGURE 3.2 – Les onze polyominos arbre d'aire 5 à symétrie près

Une énumération de tous les polyominos arbre de taille 2, 3 et 4 permet de trouver les premières valeurs de L_{\square} . Ces objets sont dessinés dans la figure 3.3. Les premières valeurs de la fonction L_{\square} sont donc,

$$L_{\square}(2) = 2, \quad L_{\square}(3) = 2, \quad L_{\square}(4) = 3, \quad L_{\square}(5) = 4.$$

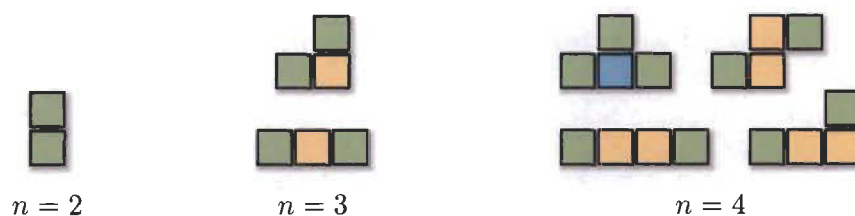


FIGURE 3.3 – Les sept polyominos arbre d'aire 2, 3 et 4

Une étude des polyominos arbre d'aire 6, 7, 8 et 9 permet de trouver des valeurs supplémentaires de la fonction L_{\square} , qui sont

$$L_{\square}(6) = 4, \quad L_{\square}(7) = 4, \quad L_{\square}(8) = 5, \quad L_{\square}(9) = 6.$$

Des polyominos arbre pleinement feuillus réalisant ces valeurs de L_{\square} sont donnés dans la figure 3.4.

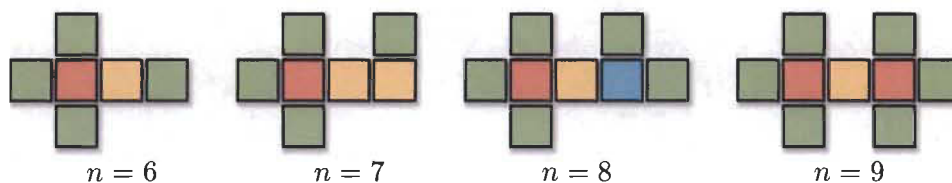


FIGURE 3.4 – Polyominos arbre pleinement feuillus d'aire 6, 7, 8 et 9

Selon les premières valeurs de L_{\square} , la fonction semble suivre un motif de croissance qui

dépend de la valeur de n modulo 4 qui est

$$L_{\square}(4k + 0) + 1 = L_{\square}(4k + 1),$$

$$L_{\square}(4k + 1) + 0 = L_{\square}(4k + 2),$$

$$L_{\square}(4k + 2) + 0 = L_{\square}(4k + 3),$$

$$L_{\square}(4k + 3) + 1 = L_{\square}(4k + 4).$$

C'est-à-dire que dépendant de la valeur de l'aire n modulo 4, en ajoutant une cellule, on peut augmenter le nombre de feuilles de 0 ou de 1. Similairement, on peut définir la valeur de $L_{\square}(n)$ en utilisant la valeur de $L_{\square}(n - 4)$. En ajoutant 4 cellules, on ajoute systématiquement 2 feuilles. C'est-à-dire que l'on a une récurrence de la forme

$$L_{\square}(n) = L_{\square}(n - 4) + 2.$$

En généralisant naïvement cette récurrence et en ajoutant les cas de base pour $n = 2, 3, 4$ et 5, on propose la fonction ℓ_{\square} , qui est une conjecture pour les valeurs de L_{\square} .

Définition 3.1.1 – Fonction ℓ_{\square}

Pour des entiers positifs n , la fonction feuille potentielle ℓ_{\square} est définie par

$$\ell_{\square}(n) = \begin{cases} 2 & \text{si } n = 2 \\ n - 1 & \text{si } n = 3, 4, 5 \\ \ell_{\square}(n - 4) + 2 & \text{sinon.} \end{cases}$$

Les premières valeurs prises par la fonction ℓ_{\square} sont données dans la table 3.1.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
ℓ_{\square}	2	2	3	4	4	4	5	6	6	6	7	8	8	8	9	10	10	10

TABLE 3.1 – Valeurs de ℓ_{\square} pour n entre 2 et 19

La périodicité dans la croissance de ℓ_{\square} se visualise plus aisément à l'aide du graphe de

$\ell_{\square}(n)$ à la figure 3.5. La courbe fait deux pas dans la direction $(1, 1)$, suivis de deux pas dans la direction $(1, 0)$.

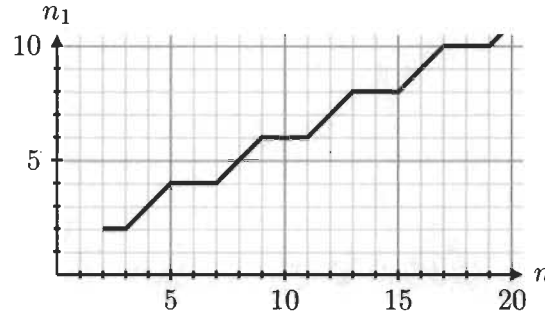


FIGURE 3.5 – Graphe de ℓ_{\square}

On montre que la fonction proposée ℓ_{\square} est bien la fonction L_{\square} . C'est-à-dire que la fonction feuille potentielle ℓ_{\square} donne le nombre maximal de feuilles d'un sous-arbre induit dans la grille carrée, qui est un polyomino arbre. Pour montrer l'égalité entre les fonctions ℓ_{\square} et L_{\square} , on montre que $\ell_{\square}(n)$ est à la fois une borne supérieure et une borne inférieure de $L_{\square}(n)$ pour toutes les valeurs de n .

Lemme 3.1.1

Pour des entiers positifs n ,

$$\ell_{\square}(n) \leq L_{\square}(n).$$

Démonstration. Pour montrer que la fonction ℓ_{\square} est bien une borne inférieure du nombre maximal de feuilles d'un polyomino arbre, on définit un ensemble de polyominos arbre F_{\square} ayant un élément d'aire n avec $\ell_{\square}(n)$ feuilles pour chaque valeur de n . Notons $F_{\square}(n)$, le polyomino arbre d'aire n que l'on construit, on a :

$$F_{\square} = \{F_{\square}(n) : n \geq 1\}.$$

Tout comme la fonction ℓ_{\square} , l'ensemble de polyominos arbre que l'on construit a 4 cas de base et une construction récursive. Les quatre polyominos arbre associés aux cas de base

sont d'aire 2, 3, 4 et 5 et ont respectivement 2, 2, 3 et 4 feuilles, ils sont illustrés dans la figure 3.6 (a), (b), (c) et (d).

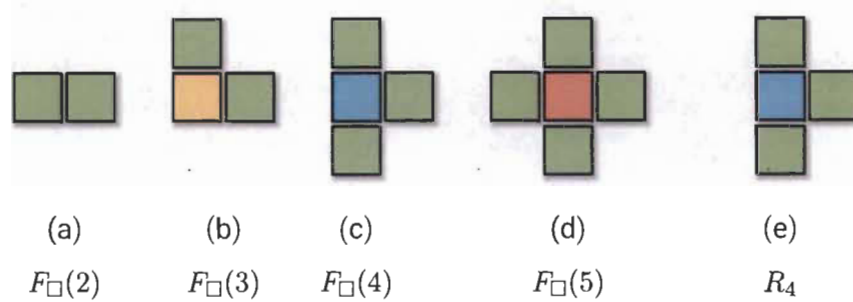


FIGURE 3.6 – Famille maximale conjecturée

Les cas de taille $n \geq 5$ sont construits en ajoutant 4 cellules à un polyomino plus petit. Considérons le polyomino dessiné à la figure 3.6 (e) R_4 .

L'ensemble de cellules ajouté R_4 a 4 cellules et 3 feuilles, mais en le juxtaposant à une feuille du cas de base, une feuille est détruite. Le nombre de feuilles de $F_{\square}(n)$ est donc 2 de plus que $F_{\square}(n - 4)$ pour un ajout de 4 cellules, ce qui est exactement la définition de ℓ_{\square} . C'est-à-dire que

$$n_1(F_{\square}(2)) = 2, \quad 3.6(a)$$

$$n_1(F_{\square}(3)) = 2, \quad 3.6(b)$$

$$n_1(F_{\square}(4)) = 3, \quad 3.6(c)$$

$$n_1(F_{\square}(5)) = 4, \quad 3.6(d)$$

$$n_1(F_{\square}(n)) = n_1(F_{\square}(n - 4)) + 2.$$

Par exemple, le polyomino arbre $F_{\square}(11)$ est construit en ajoutant deux fois l'ensemble de cellules R_4 à $F_{\square}(3)$. La construction de ce cas est donnée à la figure 3.7 (f).

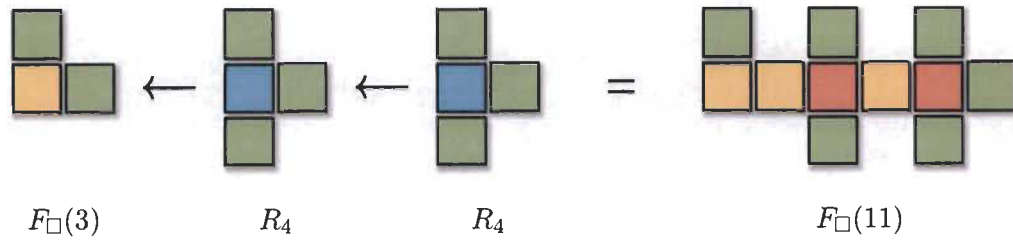


FIGURE 3.7 – Construction récursive d'un polyomino pleinement feuillu d'aire 11

□

Pour la suite, on a besoin de la fonction d'écart $\Delta_{\ell_{\square}}$ définie en 2.2.1. En particulier, dans le cas de ℓ_{\square} la fonction d'écart est définie par

$$\begin{aligned} \Delta_{\ell_{\square}}(k) &= \liminf_{n \geq k+2} \{\ell_{\square}(n) - \ell_{\square}(n-k)\} \\ &= \min_{n \geq k+2} \{\ell_{\square}(n) - \ell_{\square}(n-k)\}. \end{aligned}$$

L'utilisation de la limite inférieure n'est pas nécessaire dans le cas des polyominos, puisque la fonction feuille potentielle proposée est relativement simple. On définit donc la fonction d'écart avec le minimum plutôt que la limite inférieure. Bien que la fonction d'écart soit définie pour tout entier positif k , les seules valeurs de k qui sont utiles dans les démonstrations vont de 1 à 6.

Comme ℓ_{\square} est définie par une récurrence linéaire simple, il suffit de prendre le minimum entre quatre valeurs de n pour obtenir les valeurs de $\Delta_{\ell_{\square}}$,

$$\begin{aligned} \Delta_{\ell_{\square}}(k) &= \min \{ \ell_{\square}(8) - \ell_{\square}(8-k), \\ &\quad \ell_{\square}(9) - \ell_{\square}(9-k), \\ &\quad \ell_{\square}(10) - \ell_{\square}(10-k), \\ &\quad \ell_{\square}(11) - \ell_{\square}(11-k) \}. \end{aligned}$$

Par exemple, pour $k = 5$, on a

$$\begin{aligned}
 \Delta_{\ell_{\square}}(5) &= \min\{\ell_{\square}(8) - \ell_{\square}(8 - 5), \\
 &\quad \ell_{\square}(9) - \ell_{\square}(9 - 5), \\
 &\quad \ell_{\square}(10) - \ell_{\square}(10 - 5), \\
 &\quad \ell_{\square}(11) - \ell_{\square}(11 - 5)\}, \\
 &= \min\{5 - 2, 6 - 3, 6 - 4, 6 - 4\}, \\
 &= 2.
 \end{aligned}$$

Les valeurs de $\Delta_{\ell_{\square}}$ dont on a besoin sont données dans la table 3.2.

k	1	2	3	4	5	6
$\Delta_{\ell_{\square}}(k)$	0	0	1	2	2	2

TABLE 3.2 – Valeurs de $\Delta_{\ell_{\square}}(k)$ pour k entre 1 et 6

Le graphe de la figure 3.8 nous permet de visualiser la fonction $\Delta_{\ell_{\square}}$ comme la plus petite montée verticale de la fonction ℓ_{\square} pour un déplacement horizontal vers la droite de k unités.

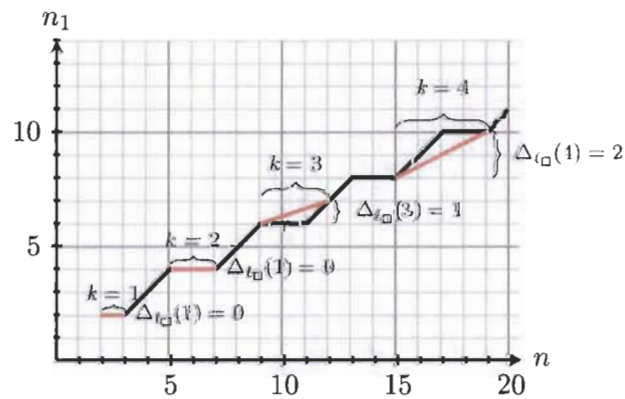


FIGURE 3.8 – Graphe de $\ell_{\square}(n)$ illustrant les premières valeurs de $\Delta_{\ell_{\square}}(k)$

Pour la démonstration du lemme 3.2.1, on utilise le lemme 2.2.1 appliqué à la fonction ℓ_{\square}

qui affirme que si deux polyominos arbre T et T' respectent les inégalités suivantes,

$$n(T') < n(T)$$

$$n_1(T) > \ell_{\square}(n(T))$$

$$n_1(T) - n_1(T') \leq \Delta_{\ell_{\square}}(n(T) - n(T')),$$

alors $n_1(T') > \ell_{\square}(n(T'))$.

3.2 Polyomino arbre enraciné figé

En spécialisant la définition 2.3.1, on obtient qu'un polyomino arbre enraciné figé est un triplet $R = (T, x, \{x, y\})$. Le triplet est formé d'un polyomino arbre T , d'une cellule x dans T appelée la racine de R et d'une arête $\{x, y\}$ dont les deux cellules x et y sont dans T . Si une arête $\{x, z\}$ est telle que x est la racine d'un polyomino arbre enraciné figé R et que z est une feuille dans R , alors l'arête $\{x, z\}$ est une direction libre de R .

On représente un polyomino arbre enraciné figé en dessinant les cellules de T , en identifiant la racine x avec un cercle et la direction en dessinant un segment allant de la cellule x à la cellule y . Un exemple de polyomino arbre enraciné figé est dessiné à la figure 3.9. On utilise l'arborescence des degrés pour décrire la structure d'un polyomino arbre enraciné figé lorsque l'on ne s'intéresse qu'à la taille et au nombre de feuilles.

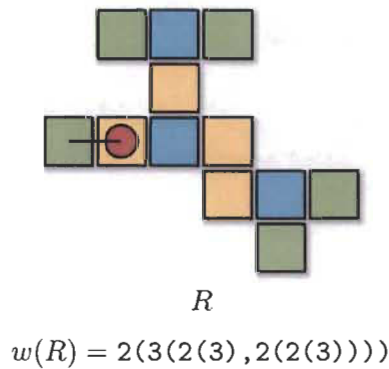


FIGURE 3.9 – Un polyomino arbre enraciné figé

Le plus petit polyomino arbre enraciné figé possible est le **polyomino arbre enraciné figé identité** de taille 2, noté **e**. Cet objet est formé de deux cellules dont l'une est la racine,

$$e = (\{x, y\}, x, \{x, y\}).$$

Il existe quatre polyominos arbre enraciné figés identités à translation près, ils sont illustrés à la figure 3.10.

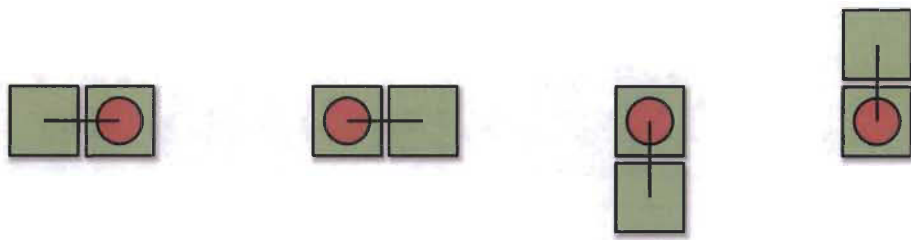


FIGURE 3.10 – Les quatre polyominos arbre enracinés figés identité à translation près

On adapte la greffe de sous-arbres induits enracinés figés 2.3.3 aux polyominos en utilisant uniquement des translations. Soit $R = (T, x, \{x, y\})$ et $R' = (T', x', \{x', y'\})$ deux polyominos arbre enracinés figés. Soit τ , la translation déplaçant la cellule y' sur la cellule x . Si la cellule $z = \tau(x')$ est telle que $\{x, z\}$ est une direction libre de R , et que l'ensemble de cellules $T \cup \tau(T')$ est une polyomino arbre, alors la greffe de R' sur R est définie par

$$R \triangleleft R' = (T \cup \tau(T'), x, \{x, y\}).$$

Un exemple de greffe bien définie est donné à la figure 3.11, alors que la figure 3.12 donne un exemple de greffe qui n'est pas bien définie, parce que l'ensemble de cellules obtenu contient un cycle et n'est donc pas un polyomino arbre enraciné figé.

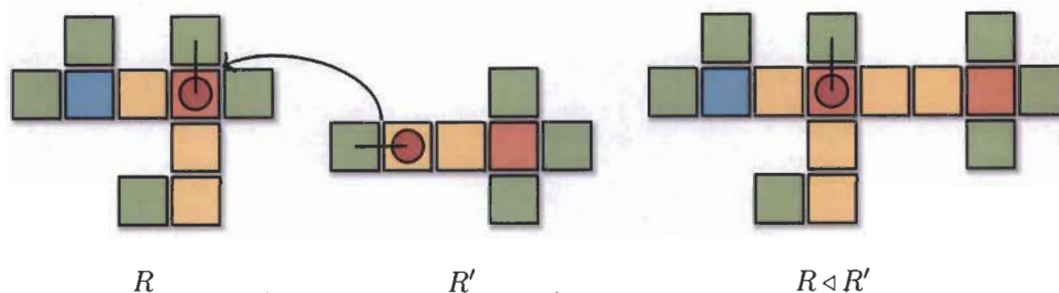


FIGURE 3.11 – Greffe bien définie entre deux polyominos arbre enracinés figés

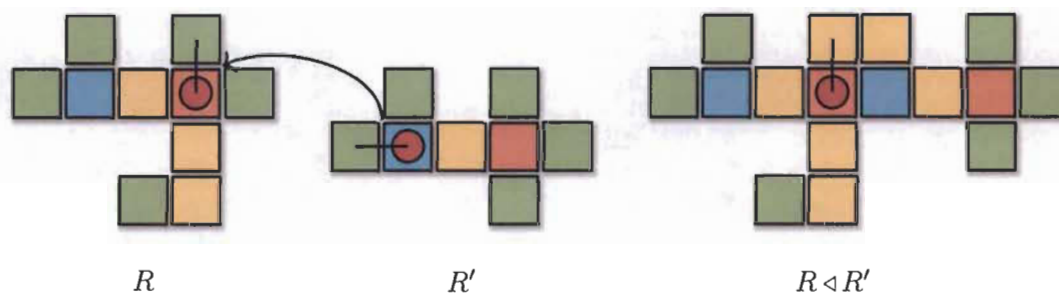


FIGURE 3.12 – Greffe non définie entre deux polyominos arbre enracinés figés

Le polyomino arbre enraciné figé identité est aussi l'élément neutre de la greffe de polyominos arbre enracinés figés. Un exemple de la greffe de e à un polyomino arbre enraciné figé est donné à la figure 3.13.

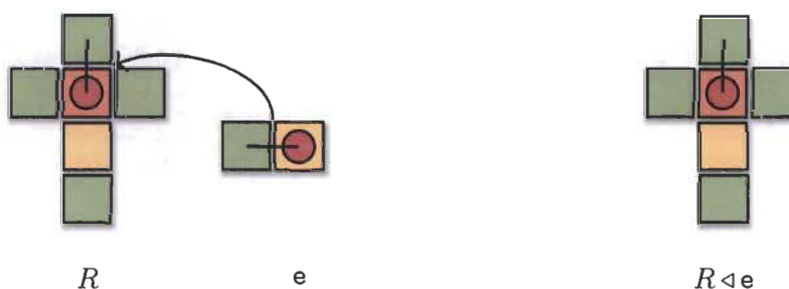


FIGURE 3.13 – Greffe du polyomino arbre enraciné figé identité

Un polyomino arbre enraciné figé R est une branche d'un polyomino P s'il existe un polyomino enraciné figé R^c appelé la co-branche de R dans P tel que $P = R \triangleleft R^c$, comme défini en 2.3.4. Une paire branche, co-branche peut être obtenue dans un polyomino P en choisissant deux sommets x et y dans P . Notons V_x et V_y , les ensembles connexes de sommets contenant respectivement x et y tels que V_x et V_y forment une partition P . Le polyomino arbre enraciné figé $R = (V_x \cup \{y\}, x, \{x, y\})$ est une branche de P et $R^c = (V_y \cup \{x\}, y, \{x, y\})$ est sa co-branche.

La figure 3.14 donne un exemple de la factorisation d'un polyomino arbre en une paire branche, co-branche.

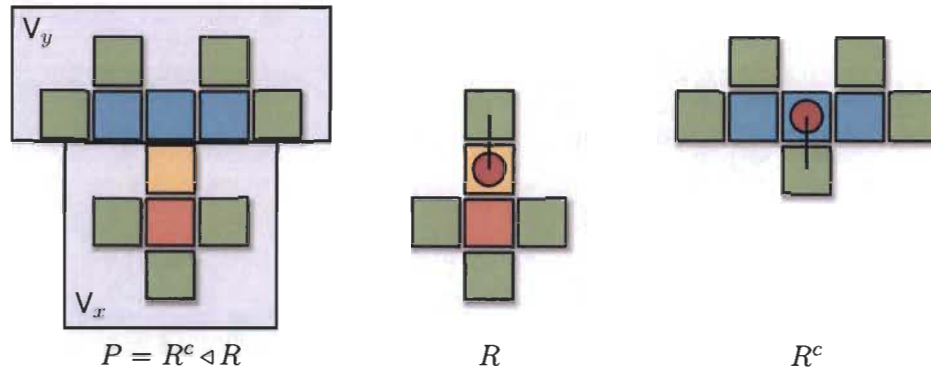


FIGURE 3.14 – Exemple de branche dans un polyomino arbre

On adapte aussi la substitution 2.3.5 aux polyominos. Soit R , une branche dans un polyomino arbre P et sa co-branche R^c . Soit R' , un polyomino arbre enraciné figé. La substitution de R par R' dans P , notée $P' = P \triangleleft_R R'$, est le premier élément du triplet dans la greffe de R' sur R^c ,

$$R^c \triangleleft R = (P', x, \{x, y\}),$$

$$P \triangleleft_R R' = P'.$$

Un exemple de substitution est donné à la figure 3.15.

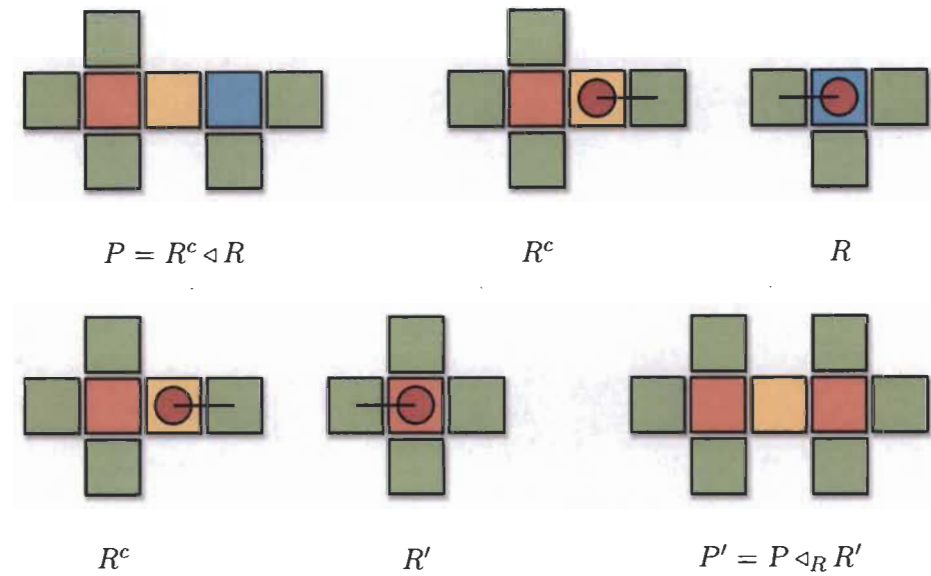


FIGURE 3.15 – Substitution dans un polyomino

Si, pour un polyomino arbre enraciné figé R , la substitution de R par un polyomino arbre enraciné figé R' est toujours bien définie, peu importe le polyomino arbre P dont R est une branche, on dit que R est substituable par R' , comme défini en 2.3.6. Rappelons qu'une condition nécessaire pour que R soit substituable par R' est que R' soit inclus dans R , lorsque l'on superpose les racines avec une translation.

On utilise un parent et un enfant d'un polyomino arbre enraciné figé à la définition 2.3.7.

Un polyomino arbre enraciné figé R est épars pour ℓ_{\square} s'il existe un autre polyomino arbre R' tel que R est substituable par R' et que

$$n_1(R) - n_1(R') \leq \Delta_{\ell_{\square}}(n(R) - n(R')).$$

Un polyomino arbre enraciné figé qui n'est pas épars pour ℓ_{\square} est abondant pour ℓ_{\square} . Un polyomino arbre enraciné figé dont tous les parents sont abondants est appelé potentiellement abondant. Ces définitions sont adaptées de la définition 2.3.8.

Lemme 3.2.1

Tous les polyominos arbre enracinés figés de profondeur 2 ou plus sont épars.

Démonstration. À la profondeur 1, les polyominos arbre enraciné figés potentiellement abondants \mathcal{P}_1 sont les polyominos arbre enracinés figés atomiques de degré 2, 3 et 4. Ces objets sont illustrés dans la figure 3.16.

Au long de la preuve, on utilise l'arborescence des degrés pour décrire les polyominos arbre enracinés figés. Par exemple, lorsque l'on écrit 2, on a en réalité 4 polyominos arbre enracinés figés. Cet abus est possible parce que tout ce qui nous intéresse pour décider si un polyomino arbre est abondant est sa taille et son nombre de feuilles, que l'on peut obtenir avec son arborescence de degrés. On a donc

$$B_1 = \{2, 3, 4\}.$$

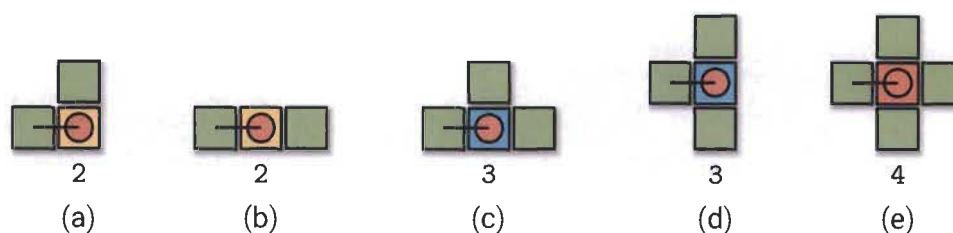


FIGURE 3.16 – Les polyominos arbre enracinés figés potentiellement abondants atomiques

Les statistiques sur ces polyominos arbre enracinés figés sont

$$n(2) = 3$$

$$n(3) = 4$$

$$n(4) = 5$$

$$n_1(2) = 2$$

$$n_1(3) = 3$$

$$n_1(4) = 4.$$

Seulement les polyominos arbre enracinés figés dont l'arborescence des degrés est 2 est épars par e , mais ceux dont l'arborescence des degrés est 3 ou 4 sont abondants,

$$\Delta(n(2) - n(e)) = 0 \geq 0 = n_1(2) - n_1(e),$$

$$\Delta(n(3) - n(e)) = 0 \not\geq 1 = n_1(3) - n_1(e),$$

$$\Delta(n(4) - n(e)) = 1 \not\geq 2 = n_1(4) - n_1(e).$$

Pour construire tous les polyominos arbre enracinés figés de profondeur 2, on greffe les polyominos arbre enracinés figés abondants de profondeur 1 sur des polyominos arbre enracinés figés atomiques de degré 2, 3 et 4. Les arborescences des degrés des polyominos arbre enracinés figés obtenus sont

$$B_2 = \{2(3), 2(4), 2(3, 3), 2(3, 4), 2(4, 4), 3(3), 3(3, 3)\}.$$

Ces objets sont illustrés dans la figure 3.17.

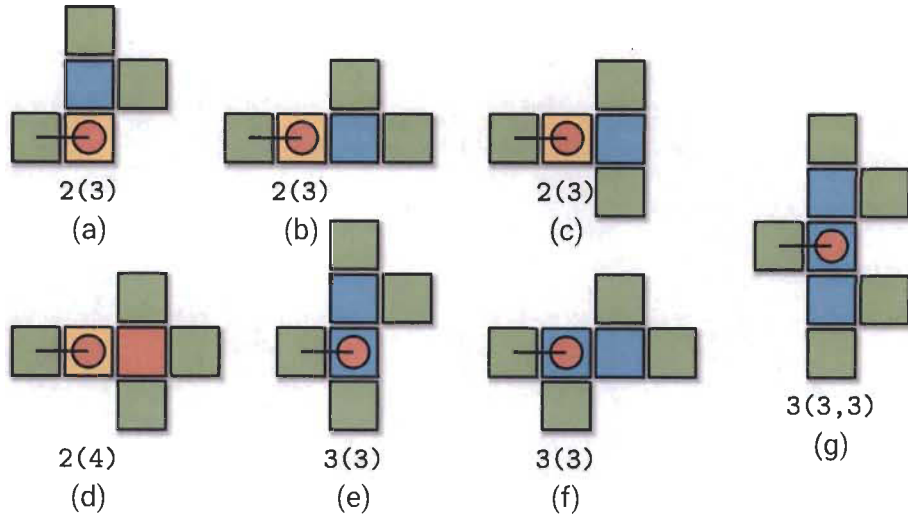


FIGURE 3.17 – Polyominos arbre enracinés figés potentiellement abondants de profondeur 2

On trouve d'abord les statistiques sur les éléments de B_2 en utilisant leur arborescence des

degrés,

$$\begin{array}{ll}
 n(2(3)) = 5 & n_1(2(3)) = 3 \\
 n(2(4)) = 6 & n_1(2(4)) = 4 \\
 n(3(3)) = 6 & n_1(3(3)) = 4 \\
 n(3(3,3)) = 8 & n_1(3(3,3)) = 5.
 \end{array}$$

On trouve que tous ces objets sont épars par \mathbf{e} , donc que $A_2 = \emptyset$,

$$\begin{aligned}
 \Delta(n(2(3)) - n(\mathbf{e})) &= 1 \leq 1 = n(2(3)) - n(\mathbf{e}) \\
 \Delta(n(2(4)) - n(\mathbf{e})) &= 2 \leq 2 = n(2(4)) - n(\mathbf{e}) \\
 \Delta(n(3(3)) - n(\mathbf{e})) &= 2 \leq 2 = n(3(3)) - n(\mathbf{e}) \\
 \Delta(n(3(3,3)) - n(\mathbf{e})) &= 2 \leq 3 = n(3(3,3)) - n(\mathbf{e}).
 \end{aligned}$$

□

On en vient alors au lemme suivant, qui est nécessaire pour le résultat principal de la section.

Lemme 3.2.2

Pour tout entier positif $n \geq 2$,

$$\ell_{\square}(n) \geq L_{\square}(n).$$

Démonstration. Supposons le contraire, c'est-à-dire qu'il existe des valeurs de n pour lesquelles

$$\ell_{\square}(n) < L_{\square}(n).$$

Choisissons m , le plus petit entier respectant cette inégalité. Il existe donc un polyomino arbre T tel que

$$\begin{aligned}
 n(T) &= m, \\
 n_1(T) &> \ell_{\square}(n(T)).
 \end{aligned}$$

Si T a au moins une branche de profondeur 2 ou une branche éparse de profondeur 1, alors T a une branche éparse, disons le polyomino arbre enraciné figé R épars par R' . Par le lemme 2.3.1, il existe un arbre $T' = T \triangleleft_R R'$ tel que

$$n(T') < n(T),$$

$$n_1(T') > \ell_{\square}(n(T')).$$

On contredit alors la minimalité de T et l'existence de l'entier m .

Si T n'a pas de branche de profondeur 2 et pas de branche éparse, on énumère tous les polyominos arbre dont la profondeur des branches est bornée par 1 et dont toutes les branches sont abondantes. Ces polyominos arbre sont illustrés dans la figure 3.18. On vérifie que chacun de ces polyominos P de taille $n(P)$ n'a pas plus de feuilles que $\ell_{\square}(n(P))$.

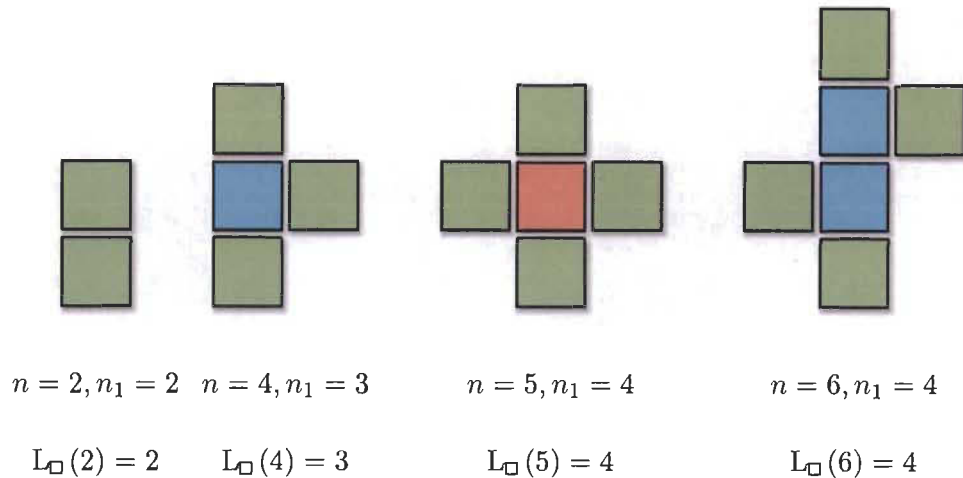


FIGURE 3.18 – Polyominos arbre dont les branches sont des polyominos arbre enracinés figés abondants dont la profondeur est au plus 1 à symétrie près

□

Proposition 3.2.1

Pour des entiers positifs n , on a $\ell_{\square}(n) = L_{\square}(n)$ et sa croissance asymptotique est donnée par

$$L_{\square}(n) \sim \frac{1}{2}n,$$

c'est-à-dire que lorsque la valeur de n devient grande, $L_{\square}(n)$ tend vers $\frac{1}{2}n$.

Démonstration. Direct des lemmes 3.1.1 et 3.2.2. □

La fonction L_{\square} peut être séparée en quatre fonctions linéaires qui sont plus utiles dans certains cas

$$L_{\square}(n) = \begin{cases} \frac{n+2}{2} & \text{si } n \equiv 0 \pmod{4}, \\ \frac{n+3}{2} & \text{si } n \equiv 1 \pmod{4}, \\ \frac{n+2}{2} & \text{si } n \equiv 2 \pmod{4}, \\ \frac{n+1}{2} & \text{si } n \equiv 3 \pmod{4}. \end{cases}$$

Similairement, $L_{\square}(n)$ peut-être définie à partir de $L_{\square}(n-1)$,

$$L_{\square}(2) = 2$$

$$L_{\square}(n) = L_{\square}(n-1) + \chi(n \equiv 0, 1 \pmod{4}).$$

On dit qu'un polyomino arbre T est **pleinement feuillu** si

$$n_1(T) = L_{\square}(n(T)) = \ell_{\square}(n(T)),$$

donc s'il a un nombre maximal de feuilles pour son aire.

3.3 Saturation

Comme la croissance de L_{\square} n'est pas régulière, certains polyominos arbre pleinement feuillus ont proportionnellement plus de feuilles que d'autres. Il est possible de caractériser certaines

familles de polyominos arbre pleinement feuillus. En particulier, pour certaines aires les polyominos arbre pleinement feuillus ont un ratio du nombre de feuilles sur le nombre de cellules plus élevé et ont ainsi une structure plus contrainte.

Remarquons d'abord que l'on peut borner la fonction L_{\square} par deux fonctions linéaires de pente $1/2$ en appliquant la proposition 2.4.1,

$$\underline{L}_{\square}(n) \leq L_{\square}(n) \leq \overline{L}_{\square}(n), \quad \forall n \geq 2.$$

Ces fonctions sont

$$\begin{aligned} \overline{L}_{\square}(n) &= \frac{n+3}{2}, \\ \underline{L}_{\square}(n) &= \frac{n+1}{2}, \end{aligned}$$

et elles sont égales à L_{\square} pour certaines valeurs de n qui sont, avec $k \geq 0$,

$$\overline{L}_{\square}(4k+1) = L_{\square}(4k+1), \quad (3.1)$$

$$\underline{L}_{\square}(4k+3) = L_{\square}(4k+3).$$

Ces fonctions sont illustrées à la figure 3.19, ce sont les deux droites au-dessus et en-dessous de la fonction L_{\square} .

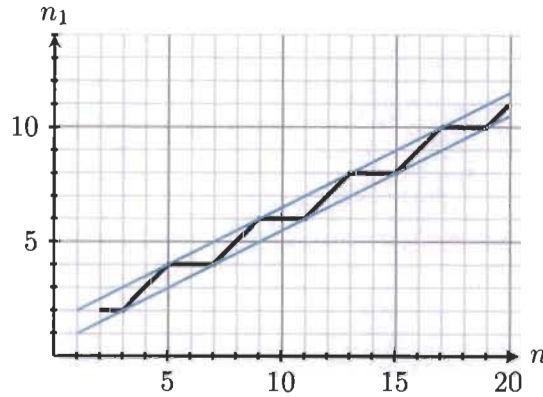


FIGURE 3.19 – Graphe des bornes de la fonction L_{\square}

De la définition 2.4.1, un polyomino arbre pleinement feuillu T est saturé s'il est tel que

$$n_1(T) = \overline{L}_{\square}(n(T)).$$

Autrement, un polyomino arbre pleinement feuillu est non saturé. Comme T est pleinement feuillu, on a

$$n_1(T) = L_{\square}(n(T)) = \overline{L_{\square}}(n(T)),$$

et donc $n(T) = 4k + 1$ par 3.1 pour un certain entier k . De plus,

$$n_1(T) = \overline{L_{\square}}(4k + 1) = 2k + 2.$$

C'est-à-dire que les polyominos arbre saturés sont d'aire $4k + 1$ et ont $2k + 2$ feuilles pour un entier $k \geq 1$. Comme les polyominos arbre saturés ont une structure leur donnant un ratio maximal du nombre de feuilles sur nombre de cellules, leur structure géométrique est plus restreinte.

De la définition 2.4.3 La fonction d'écart saturé $\sigma_{L_{\square}}$ des polyominos est défini pour tout entier positif m par

$$\sigma_{L_{\square}}(m) = L_{\square}(4k + 1) - L_{\square}(4k + 1 - m),$$

peu importe l'entier positif $k > 0$ tel que $4k + 1 > m + 2$. Les premières valeurs de la fonction $\sigma_{L_{\square}}(m)$ sont données dans la table 3.3.

m	1	2	3	4	5	6	7	8	9	10
$\sigma_{L_{\square}}(m)$	1	2	2	2	3	4	4	4	5	6

TABLE 3.3 – Valeurs de $\sigma_{L_{\square}}(m)$ pour m de 1 à 6

On adapte les définitions 2.4.2, 2.4.4 et 2.4.5 aux polyominos. Un polyomino arbre enraciné figé R est saturé s'il existe un polyomino arbre P saturé contenant R comme branche. Un polyomino arbre enraciné figé R est potentiellement saturé si pour tout polyomino arbre enraciné figé R' tel que R est substituable par R' , on a

$$n_1(R) - n_1(R') < \sigma_{L_{\square}}(n(R) - n(R')).$$

Tous les enfants d'un polyomino arbre enraciné figé potentiellement saturé sont potentiellement saturés. Être potentiellement saturé est donc une condition nécessaire pour être saturé.

Un polyomino arbre enraciné figé R est récursivement saturé s'il est potentiellement saturé et s'il existe un polyomino arbre enraciné figé R' tel que R est substituable par R' et que si un polyomino arbre saturé T contient R comme branche, alors $T' = T \triangleleft_R R'$ est aussi saturé. C'est-à-dire que R est récursivement saturé par R' s'il est potentiellement saturé et s'il existe un entier k pour lequel

$$n(R) - n(R') = 4k,$$

$$n_1(R) - n_1(R') = 2k.$$

Proposition 3.3.1

Soit T , un polyomino arbre saturé. Toutes les branches de T de profondeur paire sont de degré 2 et toutes ses branches de profondeur impaire sont de degré 4.

Démonstration. On montre cette proposition en énumérant les polyominos arbre enracinés figés potentiellement saturés de la grille carrée. À la profondeur 1, il existe 3 polyominos arbre enracinés figés atomiques dont les arborescence des degrés sont,

$$2, 3, 4.$$

En substituant ces polyominos arbre enracinés figés par \mathbf{e} , on trouve en utilisant les arborescences des degrés

$$\sigma_{\ell_{\square}}(n(2) - n(\mathbf{e})) = 1 \not\leq 0 = n(2) - n(\mathbf{e})$$

$$\sigma_{\ell_{\square}}(n(3) - n(\mathbf{e})) = 2 \not\leq 1 = n(3) - n(\mathbf{e})$$

$$\sigma_{\ell_{\square}}(n(4) - n(\mathbf{e})) = 2 \leq 2 = n(4) - n(\mathbf{e}).$$

Seulement les polyominos arbres enracinés figés dont l'arborescence des degrés est 4 sont potentiellement saturés.

À la profondeur 2, les seuls polyominos arbre enracinés figés dont tous les enfants sont potentiellement saturés sont ceux dont l'arborescence des degrés est $2(4)$. Ces polyominos arbre enracinés figés sont récursivement saturés par e ,

$$n(2(4)) - n(e) = 4$$

$$n_1(2(4)) - n_1(e) = 2 = \sigma(n(2(4)) - n(e)).$$

C'est-à-dire que dans un polyomino arbre saturé T contenant un polyomino arbre enraciné figé ayant comme arborescence des degrés $2(4)$ comme branche, on peut substituer e pour obtenir un polyomino arbre saturé T' .

Considérons un polyomino arbre saturé T . Toutes les branches R de T de profondeur 2 sont telles que $w(R) = 2(4)$. Considérons le polyomino arbre T' obtenu en substituant toutes les branches dont l'arborescence des degrés est $2(4)$ de T par e . Comme T' est saturé, toutes ses branches de profondeur 2 ont comme arborescence des degrés $2(4)$. Or, une cellule de profondeur k dans T' est de profondeur $k + 2$ dans T . C'est-à-dire que les cellules de profondeur 3 et 4 sont respectivement de degré 4 et 2 dans T .

En répétant cette procédure sur T' , on obtient le résultat désiré. Cette procédure de déconstruction est illustrée à la figure 3.20.



FIGURE 3.20 – Déconstruction récursive d'un polyomino arbre saturé

□

Définition 3.3.1 – Opérateur croix

Soit P , un polyomino arbre et n un entier positif. L'opérateur croix, $\phi : \mathcal{P}_{\square}(n) \rightarrow \mathcal{P}_{\square}(4n+1)$ est défini par

$$\phi(P) = \bigcup_{(x,y) \in P} \{(2x, 2y), (2x+1, 2y), (2x-1, 2y), (2x, 2y-1), (2x, 2y+1)\}.$$

L'opérateur croix remplace chaque cellule d'un polyomino par des croix formées de 5 cellules.

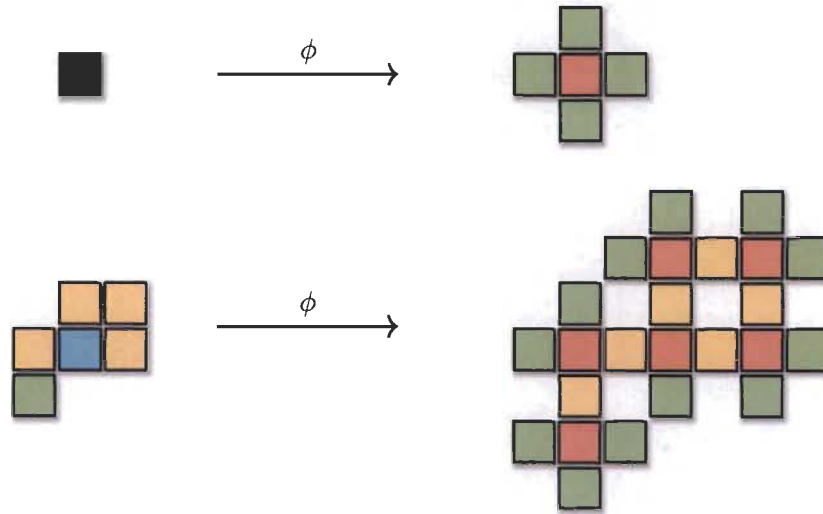


FIGURE 3.21 – Application de l'opérateur croix ϕ sur un polyomino

On s'intéresse en particulier à l'application de l'opérateur croix sur des arbre.

Proposition 3.3.2

Soit P , un polyomino et soit x et y , deux cellules adjacentes dans P . L'image de x et y dans $\phi(P)$ est connexe, c'est-à-dire que les croix associées à x et à y sont adjacentes.

Démonstration. On montre cette proposition en montrant que l'opérateur croix préserve

la relation d'adjacence. C'est-à-dire que deux croix dans l'image d'un polyomino P sont adjacentes seulement si les cellules correspondantes dans P sont adjacentes. De ce fait, on déduit que les cycles sont préservés sous l'opérateur croix.

On dit que deux croix sont adjacentes si elles partagent une cellule. Remarquons aussi qu'il ne peut y avoir de cycle entre seulement deux croix adjacentes.

Considérons deux cellules adjacentes dans le domaine de ϕ , disons (x_1, y_1) et (x_2, y_2) . Comme ces cellules sont adjacentes, elles sont à distance 1. Sans perte de généralité, posons $x_2 = x_1 + 1$ et $y_2 = y_1$.

L'image de ces deux cellules est

$$\begin{aligned}\phi(x_1, y_1) &= \{(2x_1, 2y_1), (2x_1 - 1, 2y_1), (2x_1 + 1, 2y_1), \\ &\quad (2x_1 - 1, 2y_1 - 1), (2x_1, 2y_1 + 1)\}, \\ \phi(x_1 + 1, y_1) &= \{(2x_1 + 2, 2y_1), (2x_1 + 1, 2y_1), (2x_1 + 3, 2y_1), \\ &\quad (2x_1 + 1, 2y_1 - 1), (2x_1 + 2, 2y_1 + 1)\}.\end{aligned}$$

Comme illustré dans la figure 3.22, ces deux croix sont adjacentes.

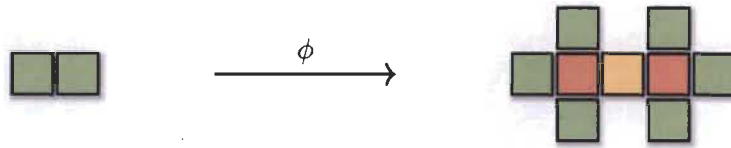


FIGURE 3.22 – Image de deux cellules adjacentes sous l'opérateur croix ϕ

□

Corollaire 3.3.1

L'image d'un polyomino arbre sous l'opérateur croix ϕ est un polyomino arbre.

Démonstration. Ce résultat est une conséquence directe de la proposition précédente.

Comme l'opérateur croix préserve l'adjacence, il préserve aussi les cycles. L'image d'un polyomino ayant 0 cycle a donc 0 cycle. \square

Dans le cas où l'opérateur croix est appliqué à un arbre, l'application peut être vue comme la greffe du polyomino arbre enraciné figé dont l'arborescence des degrés est $2(4)$ dans la direction des cellules de l'arbre. C'est-à-dire que dans un arbre T , on choisit une cellule que l'on remplace par une croix. On greffe ensuite à la cellule des polyominos arbre enracinés figés dont l'arborescence des degrés est $2(4)$ en calquant la structure de T . Cette procédure est illustrée à la figure 3.23.

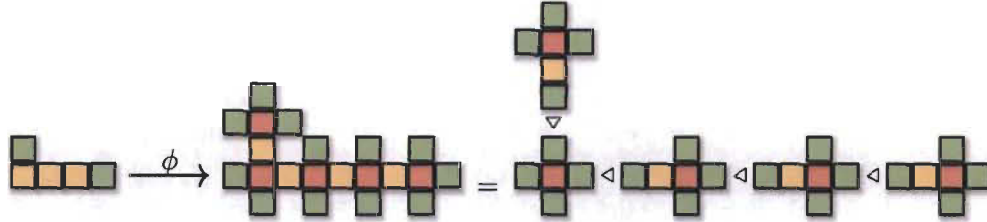


FIGURE 3.23 – Opérateur croix vu comme la greffe de croix

Comme l'application de l'opérateur croix est équivalente à la greffe de polyominos arbre enracinés figés dont l'arborescence des degrés est $2(4)$, on obtient directement le nombre de cellules et de feuilles de l'image d'un arbre sous l'opérateur croix ϕ . Comme chaque cellule correspond à la greffe d'un polyomino arbre enraciné figé dont l'arborescence des degrés est $2(4)$ ajoutant 4 cellules et deux feuilles, on a

$$n(\phi(T)) = 4(n(T)) + 1,$$

$$n_1(\phi(T)) = 2(n(T)) + 2.$$

L'image d'un arbre sous l'opérateur croix est donc un arbre saturé. En effet, il a le bon nombre de feuilles et la bonne aire,

$$n(\phi(T)) = 4k + 1,$$

$$n_1(\phi(T)) = 2k + 2 = L_{\boxtimes}(4k + 1),$$

avec $k = n(T)$.

Comme l'image d'un arbre sous l'opérateur croix est un arbre, il est possible d'itérer l'application de la fonction ϕ . En particulier, en débutant avec une seule cellule et en appliquant l'opérateur croix successivement, on obtient les structures de la figure 3.24. Notons C_0 , le polyomino arbre formé d'une seule cellule et C_k le polyomino obtenu en appliquant k fois l'opérateur croix ϕ à C_0 ,

$$C_k = \phi^k(C_0).$$

En utilisant les propriétés de l'opérateur croix, on a

$$n(C_k) = 4n(C_{k-1}) + 1,$$

$$n_1(C_k) = 2n(C_{k-1}) + 2.$$

En appliquant cette égalité successivement, on obtient les égalités suivantes,

$$n(C_k) = \frac{4^{k+1} - 1}{3},$$

$$n_1(C_k) = 3 \cdot 2^{k+1} - 2.$$

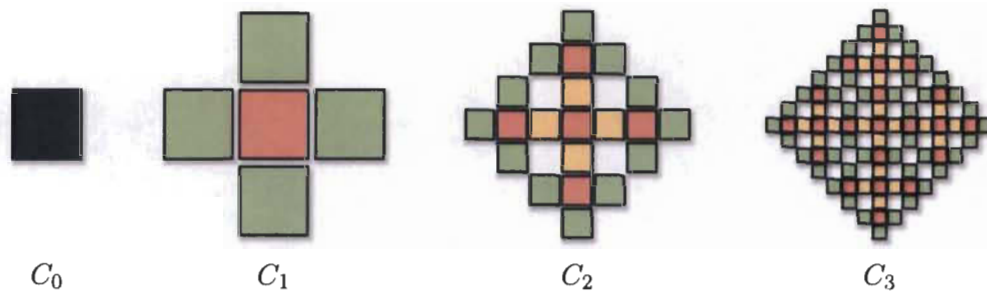


FIGURE 3.24 – Application répétée de l'opérateur croix

Proposition 3.3.3

L'opérateur croix est une bijection entre les polyominoes arbre de taille k et les polyominoes saturés de taille $4k + 1$.

Démonstration. Tout arbre saturé peut se construire en greffant successivement un polyomino arbre enraciné figé récursivement saturé dont l'arborescence des degrés est $2(4)$. On peut donc obtenir tout polyomino arbre saturé en appliquant l'opérateur croix sur un arbre, puisque l'opérateur croix est équivalent à la greffe de croix. On obtient donc qu'il y a autant de polyominos arbre saturés de taille $4k + 1$ que de polyominos arbre de taille k . C'est-à-dire que l'opérateur croix est une bijection entre les polyominos arbre de taille k et les polyominos arbre saturés de taille $4k + 1$. \square

On ne connaît par contre pas le nombre de polyominos arbre de taille donnée ce qui limite notre énumération. Notons $fl_{\square_t}(n)$ le nombre de polyominos arbre pleinement feuillus d'aire n . Pour des entiers k croissants, les premières valeurs $fl_{\square_t}(4k + 1)$, donnant simultanément le nombre de polyominos arbre d'aire k et de polyominos arbre saturés d'aire $4k + 1$, sont données dans la table 3.4. Ces suites n'apparaissent pas dans l'OEIS. Elles ont été trouvées par génération informatique.

Pour les polyominos pleinement feuillus non saturés, c'est-à-dire ceux d'aire $4k, 4k + 2$ et $4k + 3$, aucune description combinatoire n'est connue. On ne connaît aucune caractérisation efficace de ces objets permettant de les énumérer. Une génération informatique permet d'obtenir les premières valeurs de $fl_{\square_t}(4k)$, $fl_{\square_t}(4k + 2)$ et $fl_{\square_t}(4k + 3)$, qui sont données dans la table 3.4.

n	0	1	2	3	4	5	6	7	
$fl_{\square_t}(4n)$	-	4	16	72	292	1128	4324	16652	64120
$fl_{\square_t}(4n + 1)$	1	1	2	6	18	55	174	570	1908
$fl_{\square_t}(4n + 2)$	2	8	34	144	562	2144	8198	31464	-
$fl_{\square_t}(4n + 3)$	6	64	506	3070	16000	77480	361150	1637060	-

TABLE 3.4 – Nombre de polyominos arbre pleinement feuillus à translation près

Chapitre 4

Polyhexes et polytriangles

[Réf. nécessaire]

— Wikipédia

Dans ce chapitre, on se concentre sur la fonction feuille des polyhexes et des polytriangles, en particulier on conjecture cette fonction feuille à la section 4.1 et on la montre à la section 4.2. On étudie ensuite les polyhexes et les polytriangles saturés à la section 4.3. Dans le cas précis de ces grilles, il est possible d'énumérer les polyhexes et les polytriangles saturés, ce travail est fait à la section 4.4.

Remarquons que le degré maximal d'une cellule dans un polyhexe est 6, mais que si l'on contraint le polyhexe à être un arbre, alors le degré maximal est 3. Similairement, dans un polytriangle, le degré maximal est 3, puisque les cellules ont trois côtés. Dans la figure 4.1, les 4 cellules adjacentes pouvant exister dans un polyhexe arbre sont illustrées à symétrie près.

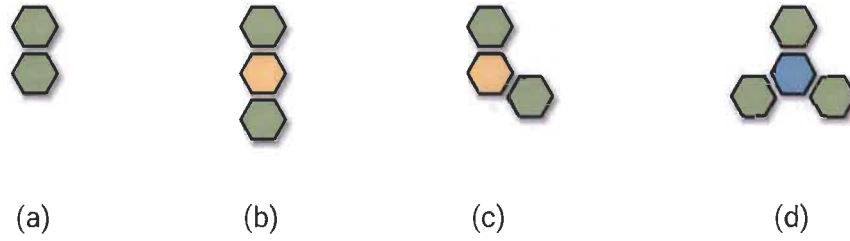


FIGURE 4.1 – Degré maximal dans un polyhexe arbre

On note respectivement $\mathcal{P}_\Delta(n)$ et $\mathcal{P}_\circ(n)$ l'ensemble des polytriangles et des polyhexes ayant n cellules. On note similairement l'ensemble des polytriangles arbre et des polyhexes arbre $\mathcal{T}_\Delta(n), \mathcal{T}_\circ(n)$.

4.1 Fonctions feuille

Afin d'étudier les arbres pleinement feuillus dans les polyhexes et les polytriangles, on adapte la définition 2.1.1 pour définir les fonctions feuille L_\circ et L_Δ avec l'égalité suivante,

$$L_\circ(n) = \max\{n_1(t) : t \in \mathcal{T}_\circ(n)\} \quad L_\Delta(n) = \max\{n_1(t) : t \in \mathcal{T}_\Delta(n)\}.$$

C'est-à-dire que les fonctions L_\circ et L_Δ donnent le nombre maximal de feuilles dans un polyhexe arbre et un polytriangle arbre. Pour trouver les valeurs de ces fonctions, on étudie tous les polyhexes arbre et les polytriangles arbre de petite taille. La figure 4.2 illustre tous les polyhexes arbre d'aire 2 (a), d'aire 3 (b), (c) et d'aire 4 (d) à (h). La figure 4.3 illustre tous les polytriangles arbre d'aire 2 (a), d'aire 3 (b) et d'aire 4 (c) à (e).

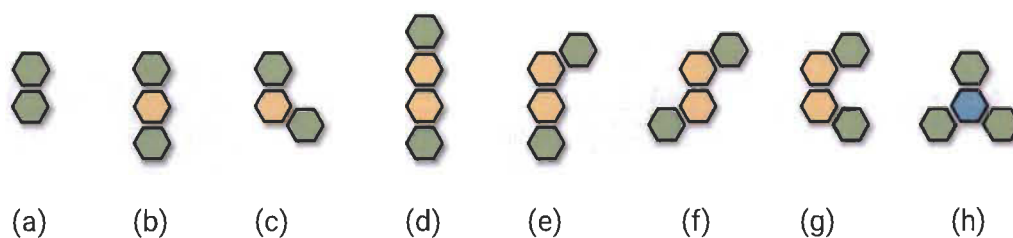


FIGURE 4.2 – Polyhexes arbre de taille 2 à 4

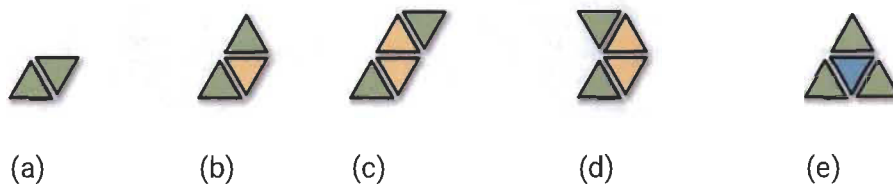


FIGURE 4.3 – Polytriangles arbre de taille 2 à 4

De plus, les figures 4.4 et 4.5 donnent des polyhexes arbre et des polytriangles arbre pleinement feuillus d'aire 5 à 8.

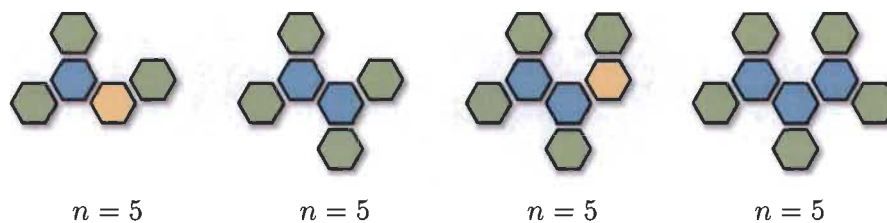


FIGURE 4.4 – Polyhexes arbre pleinement feuillus d'aire 5 à 8

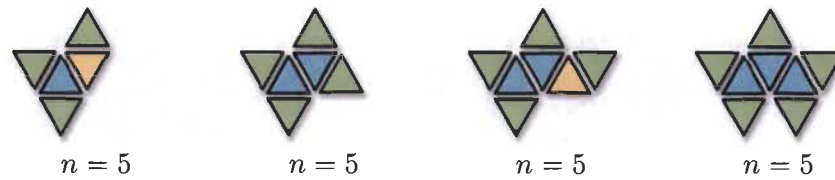


FIGURE 4.5 – Polytriangles arbre pleinement feuillus d'aire 5 à 8

De ces figures, on trouve les valeurs suivantes des fonction L_O et L_Δ ,

$$\begin{aligned} L_O(2) = L_\Delta(2) = 2, \quad L_O(3) = L_\Delta(3) = 2, \quad L_O(4) = L_\Delta(4) = 3, \quad L_O(5) = L_\Delta(5) = 3, \\ L_O(6) = L_\Delta(6) = 4, \quad L_O(7) = L_\Delta(7) = 4, \quad L_O(8) = L_\Delta(8) = 5, \quad L_O(9) = L_\Delta(9) = 5. \end{aligned}$$

Des valeurs que l'on a étudiées, c'est-à-dire les aires 2 à 8, on généralise les valeurs des fonctions feuille à tous les entiers positifs. Dans le cas des polyhexes arbre et des polytriangles arbre, les fonctions feuille semblent suivre un motif de croissance de longueur 2. C'est-à-dire qu'une fois sur deux, la valeur de la fonction feuille augmente de 1 et une fois sur deux, la valeur de la fonction feuille n'augmente pas. De façon équivalente, en augmentant la valeur de l'argument de 2, on augmente la valeur de la fonction feuille de 1.

Définition 4.1.1 – Fonctions ℓ_Δ, ℓ_O

Pour tout entier positif $n \geq 2$, les fonctions feuille potentielles ℓ_Δ et ℓ_O sont définies par

$$\ell_\Delta(n) = \ell_O(n) = \begin{cases} 2 & \text{si } n = 2, 3 \\ \ell_\Delta(n-2) + 1 & \text{sinon.} \end{cases}$$

Les fonctions feuille proposées sont égales pour les polytriangles et les polyhexes. C'est-à-dire que le nombre maximal de feuilles dans un polyhexe arbre de taille n est le même que dans un polytriangle de taille n . Afin de simplifier la notation, comme $\ell_\Delta = \ell_O$, on ne considère que ℓ_Δ lorsque les deux fonctions sont traitées.

Pour certaines démonstrations, on utilise plutôt la définition suivante, qui est équivalente,

$$\ell_{\Delta}(n) = \ell_{\square}(n) = \begin{cases} 2 & \text{si } n = 2, \\ \ell_{\Delta}(n-1) & \text{si } n \text{ est impair,} \\ \ell_{\Delta}(n-1) + 1 & \text{si } n \text{ est pair.} \end{cases}$$

Les premières valeurs de cette fonction sont données dans la table 4.1 et son graphe est donné dans la figure 4.6.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\ell_{\Delta}(n) = \ell_{\square}(n)$	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10

TABLE 4.1 – Premières valeurs de $\ell_{\Delta}(n)$ et $\ell_{\square}(n)$

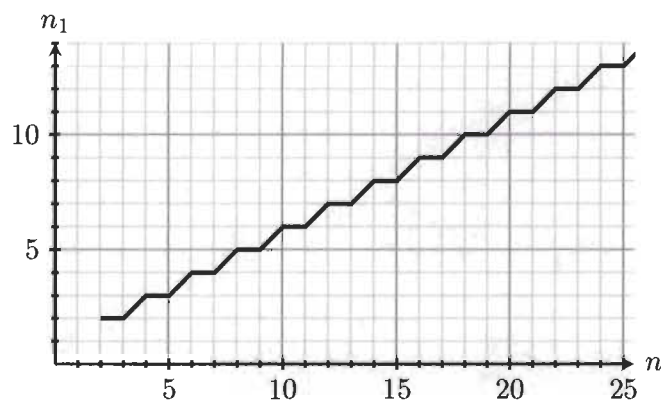


FIGURE 4.6 – Graphique de ℓ_{Δ} et ℓ_{\square}

Comme pour les polyominos, on montre l'exactitude de cette fonction en montrant qu'elle est à la fois une borne supérieure et une borne inférieure de L_{Δ} et de L_{\square} .

Lemme 4.1.1 – Borne inférieure

Pour des entiers positifs n , on a

$$\ell_{\Delta}(n) \leq L_{\Delta}(n)$$

$$\ell_{\circ}(n) \leq L_{\circ}(n).$$

Démonstration. On montre ce lemme en construisant un ensemble de polyhexes \mathcal{F}_{\circ} et un ensemble de polytriangles \mathcal{F}_{Δ} ayant chacun un élément $\mathcal{F}_{\circ}(n)$, $\mathcal{F}_{\Delta}(n)$ de taille n satisfaisant

$$n_1(\mathcal{F}_{\circ}(n)) = n_1(\mathcal{F}_{\Delta}(n)) = \ell_{\Delta}(n) = \ell_{\circ}(n)$$

pour chaque entier positif n .

Les deux ensembles sont construits similairement. L'élément de taille n est construit en ajoutant une cellule à l'élément de taille $n - 1$.

On débute avec un cas de base de taille 2 ayant 2 feuilles pour les deux ensembles, $\mathcal{F}_{\circ}(2)$ et $\mathcal{F}_{\Delta}(2)$.

Pour construire l'élément de taille n , si n est impair, on prend l'élément de taille $n - 1$ pair et on ajoute une cellule adjacente à une feuille du polyforme. La cellule ajoutée devient une feuille et la feuille sur laquelle on a ajouté une cellule devient une cellule de degré 2. On a donc une cellule de plus et aucun changement dans le nombre de feuilles, lorsque n est impair

$$n_1(\mathcal{F}_{\circ}(n)) = n_1(\mathcal{F}_{\circ}(n - 1)),$$

$$n_1(\mathcal{F}_{\Delta}(n)) = n_1(\mathcal{F}_{\Delta}(n - 1)).$$

Si n est pair, on prend l'élément d'aire $n - 1$ impair qui a une cellule de degré 2. On ajoute une cellule adjacente à la cellule de degré 2 de cet élément. La cellule ajoutée est une feuille,

ce qui augmente le nombre de cellules et le nombre de feuilles de 1.

$$n_1(\mathcal{F}_\square(n)) = n_1(\mathcal{F}_\square(n-1)) + 1,$$

$$n_1(\mathcal{F}_\triangle(n)) = n_1(\mathcal{F}_\triangle(n-1)) + 1.$$

Les cellules que l'on ajoute sont toujours dans la même direction, de façon à construire un objet linéaire. Comme la construction récursive ajoute autant de feuilles que la définition de $\ell_\triangle = \ell_\square$, on conclut que

$$n_1(\mathcal{F}_\square(n)) = n_1(\mathcal{F}_\triangle(n)) = \ell_\square(n) = \ell_\triangle(n).$$

Les premiers éléments des ensembles \mathcal{F}_\square et \mathcal{F}_\triangle sont donnés dans la figure 4.7. La construction se poursuit comme illustré pour n'importe quelle valeur de n .

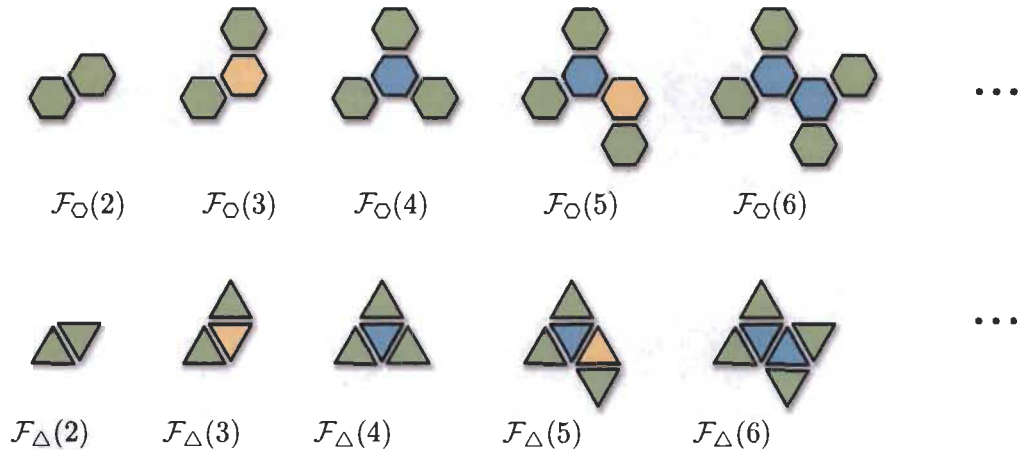


FIGURE 4.7 – Famille de polyhexes et polytriangles arbre pleinement feuillus décrite au lemme 4.1.1

□

Dans le cas des polyhexes et des polytriangles, la fonction d'écart Δ_{ℓ_\triangle} de la définition 2.2.1 devient

$$\begin{aligned} \Delta_{\ell_\triangle}(k) &= \liminf_n (\ell_\triangle(n+k) - \ell_\triangle(n)) \\ &= \min_n (\ell_\triangle(n+k) - \ell_\triangle(n)). \end{aligned}$$

L'utilisation de la limite inférieure n'est pas nécessaire puisque la fonction feuille proposée est relativement simple, on définit alors la fonction d'écart avec un minimum. Les premières valeurs de la fonction $\Delta_{\ell_{\Delta}}$ sont données dans la table 4.2.

k	1	2	3	4	5	6	7	8	9	10
$\Delta_{\ell_{\Delta}}(k)$	0	1	1	2	2	3	3	4	4	5

TABLE 4.2 – Premières valeurs de la fonction d'écart des polytriangles et polyhexes

4.2 Polyhexes et polytriangles enracinés figés

Afin d'étudier les polyhexes arbre et les polytriangles arbre pleinement feuillus, on s'intéresse à des objets plus simples qui les composent. On adapte la définition 2.3.1 aux polyhexes et aux polytriangles.

Un polyhexe (resp. polytriangle) arbre enraciné figé est un triplet $R = (T, x, \{x, y\})$ composé d'un polyhexe arbre (resp. polytriangle arbre) T , d'une cellule x dans T appelée la racine et d'une paire de cellules $\{x, y\}$ dans T appelée la direction de T .

Un polyhexe (resp. polytriangle) arbre enraciné figé $R = (T, x, \{x, y\})$ est illustré en dessinant l'arbre T , en plaçant un cercle sur la racine x et en traçant un segment entre les cellules x et y . Des exemples de polyhexes et de polytriangles enracinés figés sont illustrés dans la figure 4.8.

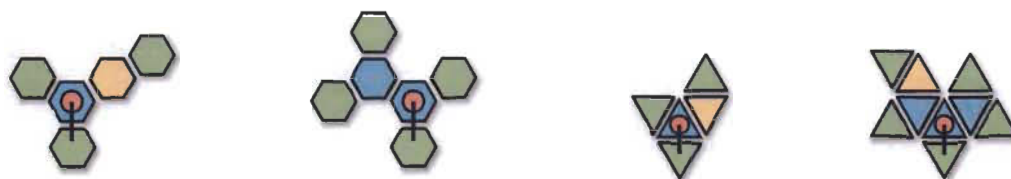


FIGURE 4.8 – Exemples de polyhexes et polytriangles arbre enracinés figés

On utilise ainsi la greffe de polyhexes et polytriangles arbre enracinés figés défini en 2.3.3 tel qu'illustré à la figure 4.9.

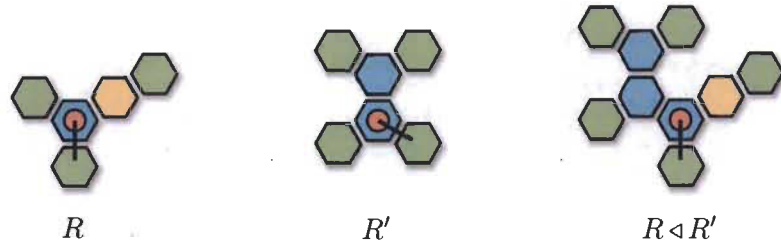


FIGURE 4.9 – Exemple d'une greffe entre deux polyhexes arbre enracinés figés

Le plus petit polyhexe (resp. polytriangle) arbre enraciné figé est le polyhexe (resp. polytriangle) arbre enraciné figé identité $e = (\{x, y\}, x, \{x, y\})$ formé de deux sommets x et y . Sa greffe sur un polyhexe (resp. polytriangle) arbre enraciné figé R redonne exactement R .

$$R \triangleleft e = R.$$

Il existe 6 polyhexes arbre enracinés figés identités et 6 polytriangles arbre enracinés figés identités à translation près, illustrés dans la figure 4.10.

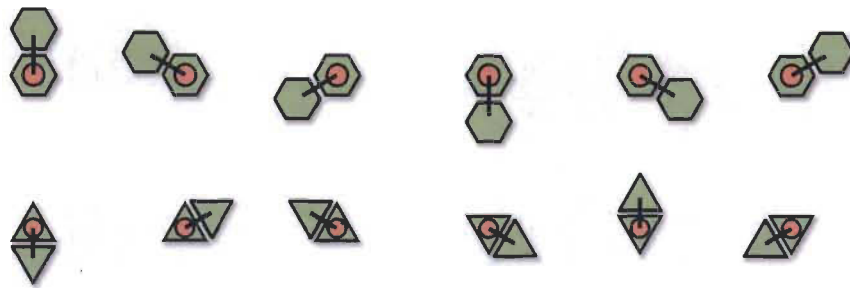


FIGURE 4.10 – Les 12 arbre enracinés figés identités

On définit aussi une branche dans un polyhexe ou un polytriangle arbre comme en 2.3.4. Un exemple d'une paire branche R , co-branche R^c dans un polyhexe arbre T est illustré à la figure 4.11.

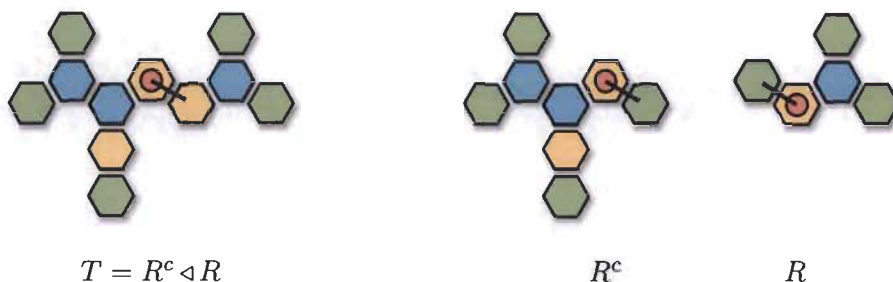


FIGURE 4.11 – Exemple d'une paire branche, co-branche dans un polyhexe arbre

Similairement, on définit la substitution d'un polyhexe (resp. polytriangle) arbre enraciné figé par un autre comme à la définition 2.3.5. Un exemple de substitution de polyhexes arbre enracinés figés est donné à la figure 4.12.

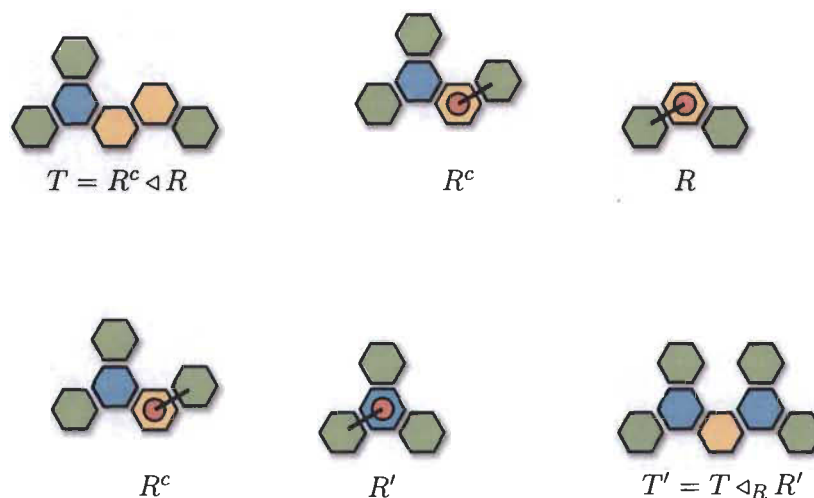


FIGURE 4.12 – Exemple de substitution dans un polyhexe

Un polyhexe (resp. polytriangle) enraciné figé R est épars s'il existe un polyhexe (resp. polytriangle) enraciné figé R' tel que R est substituable par R' et

$$n_1(R) - n_1(R') \leq \Delta_{\ell_{\Delta}}(n(R) - n(R')).$$

On dit alors que R est épars par R' pour ℓ_Δ . Si R n'est pas épars, alors il est abondant pour ℓ_Δ .

Les parents et enfants d'un polyhexe (resp. polytriangle) enraciné figé sont introduits à la définition 2.3.7. Un polyhexe (resp. polytriangle) enraciné figé est potentiellement abondant si tous ses enfants sont abondants.

Lemme 4.2.1

Tous les polyhexes (resp. polytriangles) arbre enracinés figés de profondeur 2 ou plus sont épars.

Démonstration. On commence avec les polyhexes (resp. polytriangles) arbre enracinés figés atomiques, dont les arborescences des degrés sont 2 et 3. Ces polyhexes (resp. polytriangles) sont illustrés à la figure 4.13 (a), (b), (c), (g), et (h). Le polyhexe (resp. polytriangle) enraciné figé dont l'arborescence des degrés est 2 est épars par le polyhexe (resp. polytriangle) enraciné figé identité e , mais pas celui dont l'arborescence des degrés est 3.

$$\Delta_{\ell_\Delta}(n(2) - n(e)) = 0 \geq 0 = n_1(2) - n_1(e),$$

$$\Delta_{\ell_\Delta}(n(3) - n(e)) = 0 \not\geq 1 = n_1(3) - n_1(e).$$

Le seul polyhexe (resp. polytriangle) arbre enraciné figé potentiellement abondant de profondeur 1 est donc celui dont l'arborescence des degrés est 3.

Pour construire les polyhexes (resp. polytriangles) arbre enracinés figés potentiellement abondant de profondeur 2, on greffe des polyhexes (resp. polytriangles) arbre enracinés figés abondant de profondeur 1 à un polyhexe (resp. polytriangle) arbre enraciné figé atomique. On obtient les polyhexes et polytriangles arbre enracinés figés dont les arborescences des degrés est $2(3)$, $3(3)$ ou $3(3,3)$, illustrés à la figure 4.13 qui sont tous épars par e ,

$$\Delta_{\ell_\Delta}(n(2(3)) - n(e)) = 1 \leq 1 = n_1(2(3)) - n_1(e)$$

$$\Delta_{\ell_\Delta}(n(3(3)) - n(e)) = 2 \leq 2 = n_1(3(3)) - n_1(e)$$

$$\Delta_{\ell_\Delta}(n(3(3,3)) - n(e)) = 3 \leq 3 = n_1(3(3,3)) - n_1(e).$$

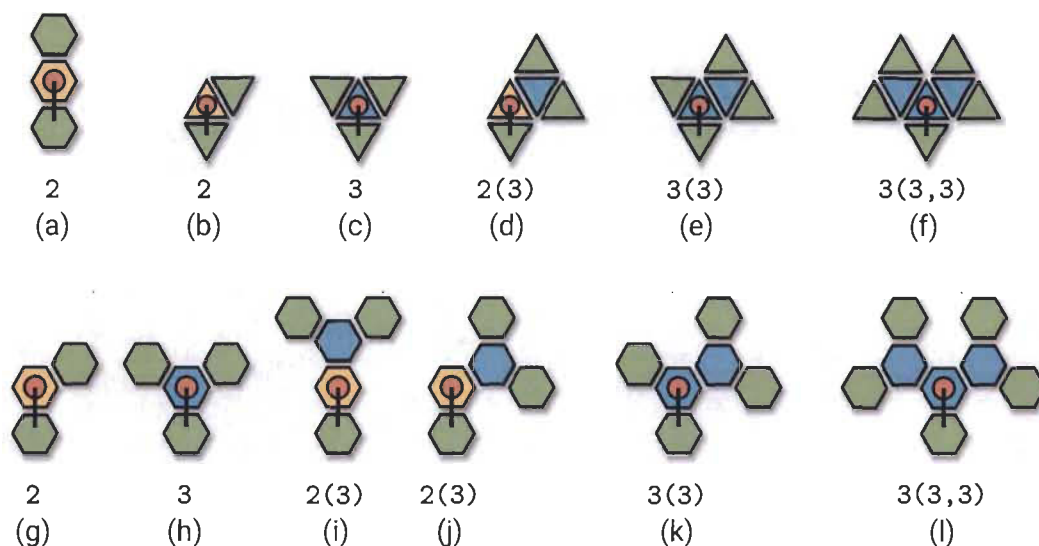


FIGURE 4.13 – Polyhexes et polytriangles arbre enracinés figés de profondeur 1 et 2

□

Lemme 4.2.2

Pour tout entier positif $n \geq 2$, on a

$$\ell_{\Delta}(n) \geq L_{\Delta}(n),$$

$$\ell_{\square}(n) \geq L_{\square}(n).$$

Démonstration. On montre cette inégalité par contradiction avec la méthode du contre-exemple minimal. Supposons que l'inégalité soit fausse et posons n le plus petit entier pour lequel l'inégalité n'est pas respectée. Soit T , un polyhexe (resp. polytriangle) arbre de taille n tel que

$$n_1(T) > \ell_{\Delta}(n(T)) = \ell_{\square}(n(T)).$$

Si T contient une branche B de profondeur 2 ou une branche éparsse de profondeur 1, alors T contient au moins un polyhexe (resp. polytriangle) arbre enraciné figé R comme branche

qui est épars. Par le lemme 3.2.1, il existe donc un arbre T' tel que

$$n(T') < n(T),$$

$$n_1(T') \geq \ell_O(n(T')) = \ell_\Delta(n(T')).$$

contredisant la minimalité et l'existence de T .

Si T n'a ni branche de profondeur 2 ni branche épars, on construit tous les polyhexes (resp. polytriangles) arbre dont la profondeur maximale des branches est bornée par 1 et dont toutes les branches sont abondantes. Ces polyhexes et polytriangles arbre sont illustrés à symétrie près à la figure 4.14. On vérifie qu'aucun de ces arbres n'a plus de feuilles que ce qui est prédit par la fonction $\ell_\Delta = \ell_O$.

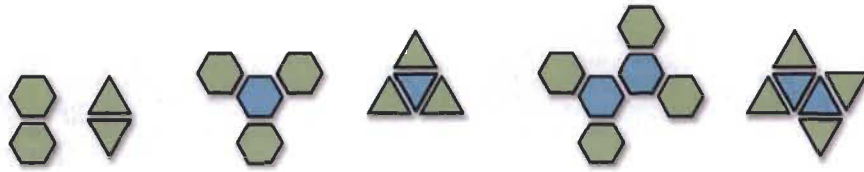


FIGURE 4.14 – Polyhexes dont les branches sont abondantes et de profondeur inférieure à 2

□

Corollaire 4.2.1

Pour des entiers positifs n , on a

$$\ell_\Delta(n) = L_\Delta(n)$$

$$\ell_O(n) = L_O(n).$$

et sa croissance asymptotique est

$$L_\Delta(n) = L_O(n) \sim \frac{1}{2}n.$$

Démonstration. Directe des lemmes 4.1.1 et 4.2.2.

□

Comme les fonctions L_Δ et L_\square sont égales, on ne mentionne que L_Δ pour la suite. Il est possible de séparer cette fonction en deux fonctions linéaires, qui sont utiles pour étudier les arbres saturés

$$L_\Delta(n) = \begin{cases} \frac{n+2}{2} & \text{si } n \equiv 0 \pmod{2}, \\ \frac{n+1}{2} & \text{si } n \equiv 1 \pmod{2}. \end{cases}$$

4.3 Saturation

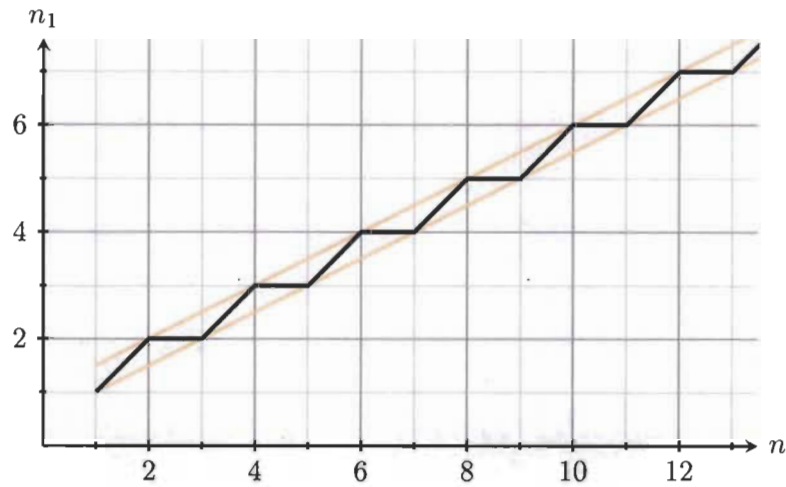
De la proposition 2.4.1, comme L_Δ est définie par une récurrence linéaire simple, il existe deux fonctions linéaires de pente $\frac{1}{2}$ qui bornent L_Δ supérieurement et inférieurement.

Ces fonctions, représentées par les droites sur la figure 4.15, sont

$$\begin{aligned} L_\Delta(n) &\geq \underline{L}_\Delta(n) = \frac{n+2}{2}, \\ L_\Delta(n) &\leq \overline{L}_\Delta(n) = \frac{n+1}{2}. \end{aligned}$$

Dans le cas des polyhexes et des polytriangles, comme L_Δ ne dépasse jamais \overline{L}_Δ , un polyhexe ou polytriangle T est saturé si

$$n_1(T) = \overline{L}_\Delta(n(T)).$$

FIGURE 4.15 – Graphe des fonctions bornant L_Δ

Les polytriangles et polyhexes saturés sont d'aire $2k$ et ont $k + 1$ feuilles pour tout entier positif $k > 0$, c'est pour ces valeurs $n = 2k$ que

$$\overline{L}_\Delta(n) = L_\Delta(n).$$

Un polyhexe (resp. polytriangle) non saturé est un polyhexe (resp. polytriangle) arbre pleinement feuillu d'aire $2k + 1$ ayant $k + 1$ feuilles. La structure des polytriangles et des polyhexes saturés est suffisamment restreinte pour pouvoir les énumérer.

De la définition 2.4.3, la fonction d'écart saturé dans ces grilles, σ_{L_Δ} devient, pour $m > 0$ et pour k entier tel que $2k > m + 1$,

$$\sigma_{L_\Delta}(m) = L_\Delta(2k) - L_\Delta(2k - m).$$

Les premières valeurs de la fonction d'écart saturé sont données dans la table 4.3.

m	1	2	3	4	5	6	7	8	9	10
$\sigma_{L_\Delta}(m)$	1	1	2	2	3	3	4	4	5	5

TABLE 4.3 – Valeurs de $\sigma_{L_\Delta}(m)$ pour m de 1 à 6

Afin de caractériser les polyhexes et les polytriangles arbre pleinement feuillus, on étudie les polyhexes et les polytriangles arbre enracinés figés saturés. Un polyhexe (resp. polytriangle) arbre enraciné figé est saturé s'il existe un polyhexe (resp. polytriangle) arbre saturé qui le contient comme branche. Comme cette condition est difficile à vérifier, on utilise plutôt une condition plus faible. Un polyhexe (resp. polytriangle) arbre enraciné figé R est potentiellement saturé si pour tout polyhexe (resp. polytriangle) arbre enraciné figé R' tel que R est substituable par R' , on a

$$n_1(R) - n_1(R') > \sigma_{L_\Delta}(n(R) - n(R')).$$

On déduit que tout polyhexe (resp. polytriangle) arbre enraciné figé saturé est potentiellement saturé.

Proposition 4.3.1

Pour tout entier positif $k \geq 1$, dans un polyhexe (resp. polytriangle) saturé, toutes les cellules internes, de profondeur k , sont de degré 3.

Démonstration. On montre cette proposition en étudiant les polyhexes et les polytriangles arbre enracinés figés potentiellement saturés. À la profondeur 1, il existe deux polyhexes (resp. polytriangles) enracinés figés atomiques, dont les arborescences des degrés sont 2 et 3. En les remplaçant par e , on trouve que ceux dont l'arborescence des degrés est 3 est potentiellement saturé,

$$\sigma_{L_\Delta}(n(2) - n(e)) = 1 \not\leq 0 = n_1(2) - n_1(e)$$

$$\sigma_{L_\Delta}(n(3) - n(e)) = 1 \leq 1 = n_1(3) - n_1(e).$$

De plus, le polyhexe (resp. polytriangle) arbre enraciné figé dont l'arborescence des degrés est 3 est récursivement saturé par e ,

$$n(3) - n(e) = 2$$

$$n_1(3) - n_1(e) = 1 = \sigma_{L_\Delta}(n(3) - n(e)).$$

C'est-à-dire que dans un polyhexe (resp. polytriangle) saturé T , toutes les branches de profondeur 1 sont de degré 3. De plus, en substituant toutes ses branches de profondeur 1

par e , on obtient un polyhexe (resp. polytriangle) arbre saturé T' . Comme T' est saturé, toutes ses branches de profondeur 1 sont de degré 3. Comme les branches de profondeur 1 dans T' sont des branches de profondeur 2 dans T , on a que toutes les branches de profondeur 2 dans T sont de degré 3. En répétant cette procédure un nombre arbitraire de fois, on trouve que toutes les branches de T sont de degré 3, donc que toutes les cellules internes sont de degré 3 \square

Similairement, il est possible de caractériser la distribution des degrés des cellules d'un polyhexe ou polytriangle arbre pleinement feuillu non saturé.

Proposition 4.3.2

Pour tout entier positif $k \geq 1$, soit T un polyhexe (resp. polytriangle) pleinement feuillu non saturé de taille $n = 2k + 1$. On a

$$\begin{aligned} n(T) &= 2k + 1, & n_1(T) &= k + 1, \\ n_2(T) &= 1, & n_3(T) &= k - 1. \end{aligned}$$

Démonstration. De la théorie des graphes [Lab81], on sait que pour un arbre T

$$\sum_{i \geq 1} in_i(T) = 2n(T) - 2.$$

Puisque T est non saturé, on a $n(T) = 2k + 1$ et $n_1(T) = k + 1$ pour un certain entier k et

$$\sum_{i > 1} in_i(T) + k + 1 = 4k,$$

et donc

$$\sum_{i > 1} in_i(T) = 3k - 1 \quad (*).$$

De plus, en sommant sur tous les degrés possibles dans un arbre quelconque, on a

$$\sum_i n_i(T) = 2k + 1.$$

En isolant le nombre de feuilles $n_1(T) = k + 1$, on obtient

$$n_2(T) + n_3(T) = k \quad (**).$$

Comme les deux seuls degrés possibles sont 2 et 3, l'équation (*) devient

$$2n_2(T) + 3n_3(T) = 3k - 1,$$

En soustrayant deux fois l'équation (**) à l'équation (*) on trouve $n_3(T) = k - 1$ et finalement $n_2(T) = 1$. \square

4.4 Énumération

On note respectivement $\text{fl}_\Delta(n)$, $\text{fl}_\square(n)$ le nombre de polytriangles et polyhexes arbre pleinement feuillus d'aire n à translation près.

Proposition 4.4.1 – Nombre de polytriangles saturés

Pour des entiers positifs k , le nombre de polytriangles arbre saturés fixés de taille $n = 2k$ est donné par

k	1	2	3	4	5	≥ 6
$\text{fl}_\Delta(2k)$	3	2	3	6	9	6

Démonstration. De la proposition 4.3.1 toutes les cellules internes d'un polytriangle arbre saturé sont de degré 3. On construit les polytriangles arbre saturés en plaçant les cellules de degré 3 et en ajoutant des feuilles autour. Pour qu'une cellule soit de degré 3, elle doit avoir un voisin sur chacun de ses 3 côtés.

Si les cellules internes sont placées en chaîne, on montre qu'elles doivent être placées en pilier, c'est-à-dire comme dans la figure 4.16 (a). Les cellules internes ne peuvent pas être placées comme à la figure 4.16 (b) sinon en ajoutant des cellules sur tous les côtés des triangles on obtient un cycle, comme à la figure 4.16 (c).

Si les cellules internes ne sont pas placées en chaîne, on doit avoir au moins 4 cellules internes dans la configuration de la figure 4.16 (d). Or il est impossible d'avoir 5 cellules internes avec cette formation, comme dans la figure 4.16 (e). L'ajout d'une autre cellule interne à 4.16 (d) impose un cycle lors de l'ajout des feuilles pour obtenir un polytriangle saturé, voir figure 4.16 (f).

Un polytriangle saturé est donc soit un pilier, soit il est d'aire 10 et n'est pas une chaîne. Si c'est un pilier, il y en a 6 à translation près dès que le nombre de cellules internes est au moins 3, voir figure 4.17 (a). Si c'est la structure exceptionnelle d'aire 10, il y en a 2 orientations à translation près, voir figure 4.17 (b).

En trouvant les cas de base, pour 0, 1 ou 2 cellules internes, on complète la démonstration. □

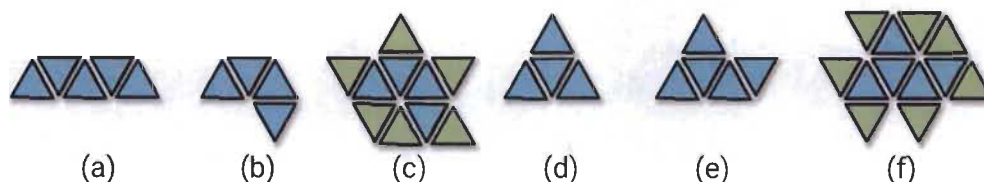


FIGURE 4.16 – Cas de la preuve de la proposition 4.4.1

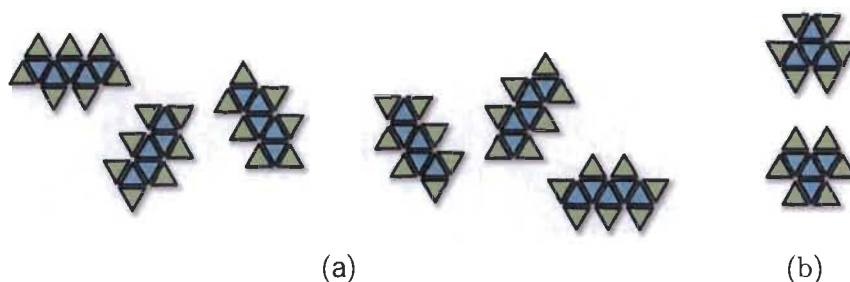


FIGURE 4.17 – Polytriangles saturés

Proposition 4.4.2 – Nombre de polyhexes saturés

Pour des entiers positifs n , le nombre de polyhexes saturés fixés d'aire n est donné par

k	1	2	3	4	5	≥ 6
$f_{\text{O}}(2k)$	3	2	3	6	9	6

Démonstration. Comme pour la démonstration de la proposition 4.4.1, on montre qu'une chaîne de cellules internes doit être placée comme dans la figure 4.18 (a). En effet, dans les autres configurations, celles de la figure 4.18 (b) et (c) les cellules ne peuvent pas être de degré 3, puisqu'en ajoutant des feuilles pour obtenir un polyhexe saturé, on crée un cycle. Dans le cas où les cellules internes ne sont pas en chaîne, on montre que la structure ayant 5 cellules internes de la figure 4.18 (d) ne peut pas exister, puisqu'encore une fois on crée un cycle en ajoutant les feuilles d'un polyhexe saturé.

On trouve donc 6 piliers différents à translation près pour chaque aire, voir figure 4.19 (a). Dans le cas $n = 10$, on trouve deux structures sporadiques à translation près, voir figure 4.19 (b). En trouvant les valeurs pour 0, 1 et 2 cellules internes, on complète la démonstration. \square

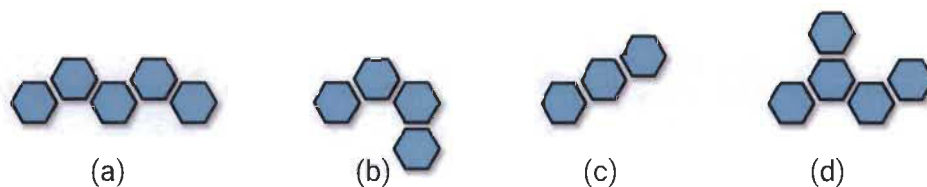


FIGURE 4.18 – Polyhexes saturés

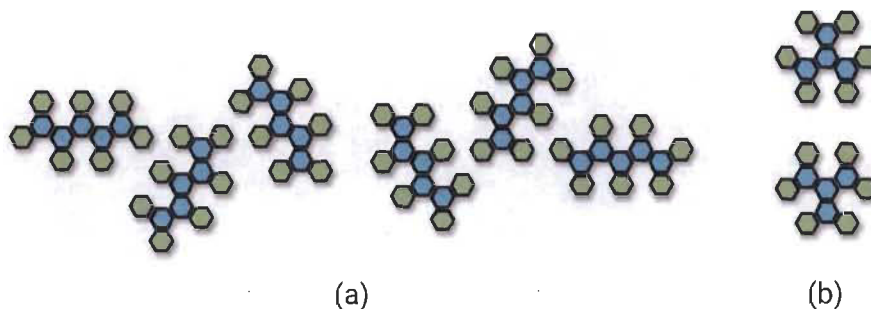


FIGURE 4.19 – Symétries des polyhexes saturés

Pour les polyhexes et les polytriangles, les arbres saturés sont équivalents à symétrie près à un exception près. On pourrait donc montrer qu'il n'existe qu'un seul polytriangle et un seul polyhexe arbre saturé pour chaque aire, sauf $n = 10$ où il en existe 2. Ce résultat est démontré dans l'article [BdG18a].

On a une égalité entre le nombre de polytriangles saturés et de polyhexes saturés à translation près, il est donc possible de construire une fonction bijective d'un ensemble à l'autre.

Définition 4.4.1

Pour tout entier $n \geq 2$, considérons la fonction $\phi : \mathcal{T}_{\Delta}(n) \rightarrow \mathcal{T}_{\square}(n)$ qui tronque les triangles d'un polytriangle fixé en hexagones pour obtenir un polyhexe fixé. Cette fonction est illustrée dans la figure 4.20.

Bien que cette fonction soit définie sur tous les polytriangles, on s'y intéresse en particulier pour les polytriangles arbre pleinement feuillus, et plus particulièrement pour la famille des polytriangles saturés.

Proposition 4.4.3

Pour chaque entier positif n , la fonction ϕ est bien définie, injective et préserve la structure de graphe.

Démonstration. Cette fonction préserve l'adjacence, c'est-à-dire que deux hexagones dans l'image d'un polytriangle sont adjacents seulement si les deux triangles correspondants dans la préimage sont adjacents. L'image d'un polytriangle est donc un polyhexe et la fonction est bien définie. En particulier, l'image d'un polytriangle arbre est un polyhexe arbre.

Comme la correspondance ne déplace pas, n'ajoute pas et ne retire pas de cellule, on constate que deux polytriangles différents ont deux images différentes, la fonction est donc injective. \square

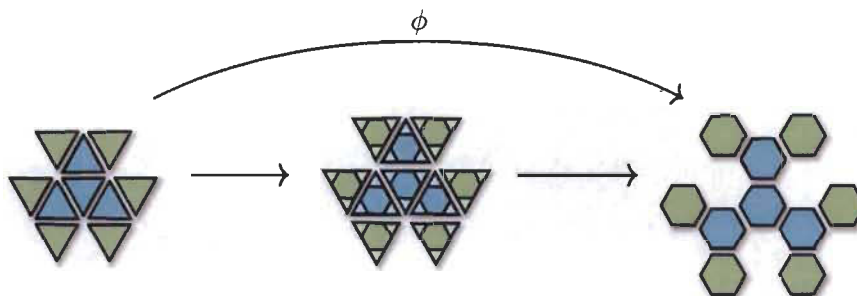


FIGURE 4.20 – Fonction ϕ entre polytriangles et polyhexes saturés

Cette correspondance n'est pas surjective en général. En effet, la figure 4.21 illustre un plus grand nombre de polyhexes formés d'une cellule de degré 2 et de deux feuilles que de polytriangles ayant la même structure. C'est-à-dire qu'en se restreignant à deux ensembles similaires dans les polytriangles et les polyhexes, il y aura au moins autant de polyhexes que de polytriangles. Par exemple, on conjecture qu'il existe strictement plus de polyhexes pleinement feuillus non saturés que de polytriangles pleinement feuillus non saturés.

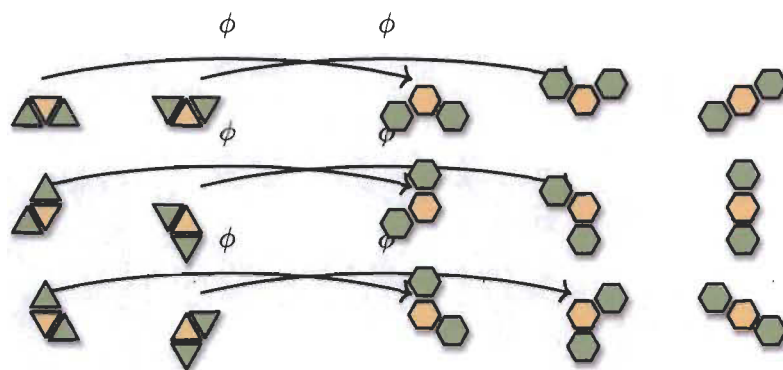


FIGURE 4.21 – Exemple de la non surjectivité de la fonction ϕ sur les polytriangles

Corollaire 4.4.1

La restriction ϕ_S de la fonction ϕ aux polytriangles arbre saturés vers les polyhexes arbre saturés est une bijection.

Démonstration. Comme la fonction ϕ est injective et que les deux ensembles ont la même cardinalité, la fonction ϕ est surjective. \square

Chapitre 5

Polycubes pleinement feuillus

Jamais deux sans trois

— Le nombre de dimensions

Similairement aux deux chapitres précédents, dans ce chapitre on s'intéresse aux polycubes pleinement feuillus. On trouve donc la fonction feuille de la grille cubique aux section 2.1 et 2.2. À la section 2.3, on étudie les polycubes arbre saturés et on conjecture une bijection permettant de les énumérer simplement.

Il est parfois difficile de visualiser dans le plan un polycube, qui est un objet en trois dimensions. Des stratégies sont utilisées pour rendre les objets plus faciles à imprimer sur papier. Lorsque la position des feuilles n'importe pas, on omet parfois de les dessiner, afin de mieux visualiser la structure interne du polycube. Le dessin à l'aide de graphes est aussi utile afin de visualiser certains polycubes ayant beaucoup de cellules. Lorsqu'un cube est caché, la coloration des cellules selon leur degré permet de deviner son existence. Par exemple, si une cellule est de degré 4 et que l'on ne voit que 3 de ses voisins, on devine qu'un de ses voisins est caché.

Dans la figure 5.1, le même polycube est dessiné de cinq façons permettant de découvrir la structure. On a donc un polycube dessiné normalement en (a), dessiné comme sous-graphe

induit de la grille cubique en (b), dessiné sans les feuilles en (c) et en (d) et dessiné avec de plus petits cubes en (e). On privilégie le plus souvent la méthode de dessin des figures (a) et (c)

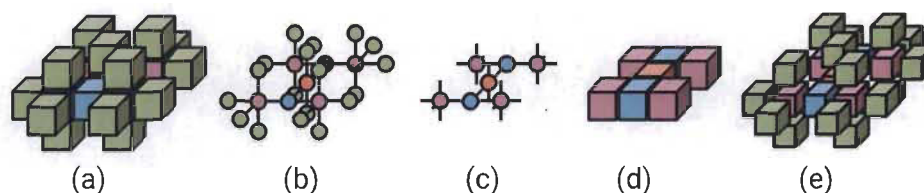


FIGURE 5.1 – Différentes façons de dessiner un polycube

Notons que pour deux cellules adjacentes dans un polycube arbre, la somme des degrés de ces deux cellules est bornée par 8. Si l'on a une cellule de degré 6, par exemple, on sait qu'aucun de ses voisins n'aura un degré supérieur à 2. Similairement, le degré maximal d'une cellule voisine à une cellule de degré 5 est 3. Les huit paires de degrés de cellules voisines sont donc

$$\begin{array}{lll} \{2, 2\}, & \{2, 3\}, & \{2, 4\}, \\ \{2, 5\}, & \{2, 6\}, & \{3, 3\}, \\ \{3, 4\}, & \{3, 5\}, & \{4, 4\}. \end{array}$$

Les paires de cellules de degrés maximaux, c'est-à-dire auxquelles on ne peut pas ajouter de feuilles sont illustrés dans la figure 5.2. Ces paires sont $\{2, 6\}$ en (a), $\{3, 5\}$ en (b) et $\{4, 4\}$, en (c).

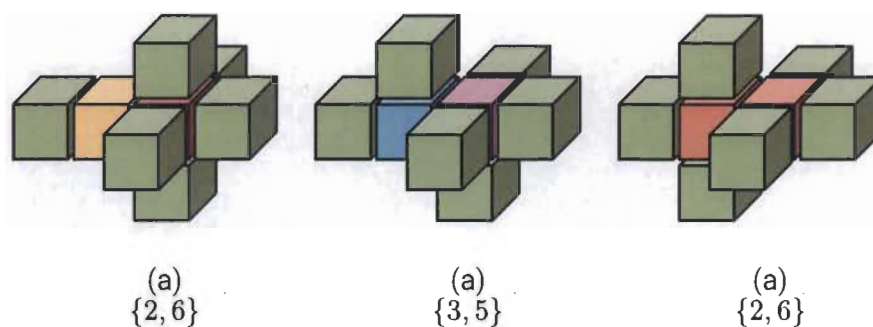


FIGURE 5.2 – Paires de cellules de degrés maximaux

Sauf lorsque mentionné, les polycubes sont considérés à translation près. Les polycubes sont habituellement dessinés à symétrie près afin d'alléger les figures. On note $\mathcal{P}_{\mathfrak{a}}(n)$ et $\mathcal{T}_{\mathfrak{a}}(n)$, respectivement l'ensemble des polycubes et des polycubes arbre de taille n .

5.1 Fonction feuille

La définition 2.1.1 est adaptée au cas des polycubes pour obtenir la fonction $L_{\mathfrak{a}}$ définie pour tout entier positif $n \geq 2$ par

$$L_{\mathfrak{a}}(n) = \max\{n_1(T) : T \in \mathcal{T}_{\mathfrak{a}}(n)\}.$$

La fonction $L_{\mathfrak{a}}$ donne le nombre maximal de feuilles que peut avoir un polycube arbre de taille n . Un polycube arbre T est dit **pleinement feuillu** s'il a un nombre maximal de feuilles pour sa taille, c'est-à-dire qu'il respecte l'égalité suivante

$$n_1(T) = L_{\mathfrak{a}}(n(T)).$$

Pour trouver les valeurs de $L_{\mathfrak{a}}$, on débute en explorant les polycubes arbre de petites tailles. Tous les polycubes arbre de taille 2 et 3 sont illustrés à symétrie près à la figure 5.3. De plus, les polycubes arbre pleinement feuillus de taille 4, 5, 6 et 7 sont illustrés à la figure 5.4.

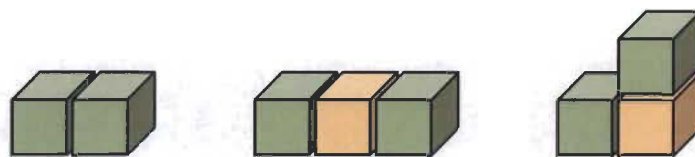


FIGURE 5.3 – Polycubes arbre de taille 2 et 3

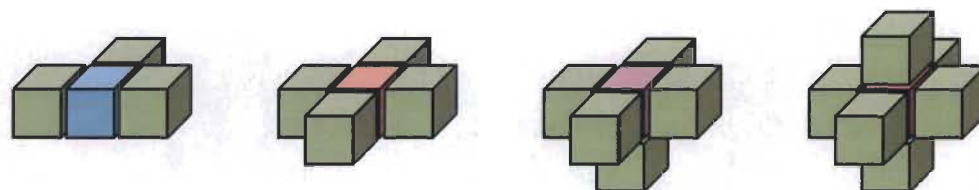


FIGURE 5.4 – Polycubes arbre pleinement feuillus de taille 4, 5, 6 et 7

On obtient ainsi les valeurs suivantes,

$$\begin{array}{lll} L_{\sigma}(2) = 2 & L_{\sigma}(3) = 2, & L_{\sigma}(4) = 3, \\ L_{\sigma}(5) = 4 & L_{\sigma}(6) = 5, & L_{\sigma}(7) = 6. \end{array}$$

Pour obtenir les valeurs de la fonction L_{σ} , on pourrait croire qu'une généralisation de la fonction feuille des polyominos fonctionnerait. En effet, la partie récursive permettant d'obtenir la famille maximale dans les polyominos est construite en ajoutant un ensemble de 4 cellules qui ajoute 2 feuilles et une structure similaire existe pour les polycubes, qui ajoute 6 cellules et 4 feuilles. Cette structure est composée d'une cellule de degré 6, d'une cellule de degré 2 et de 6 feuilles. La construction d'une conjecture pour les valeurs de L_{σ} à partir de cette idée donne la fonction l_{σ} définie par

$$l_{\sigma}(n) = \begin{cases} 2 & \text{si } n = 2 \\ n - 1 & \text{si } n = 3, 4, 5, 6, 7 \\ l_{\sigma}(n - 6) + 4 & \text{sinon.} \end{cases}$$

Les polycubes construits avec cette méthode ont une forme similaire au polycube de la figure 5.5 (a). Cette fonction est égale à L_{σ} jusqu'à la taille $n = 15$ où l'on obtient un

premier contre-exemple à l'exactitude de la fonction $l_{\mathfrak{G}}$. La méthode de construction d'un polycube de taille 15 imitant les polyominos pleinement feuillus donne 10 feuilles alors que l'arbre illustré à la figure 5.5 (b) a aussi 15 cellules mais 11 feuilles.

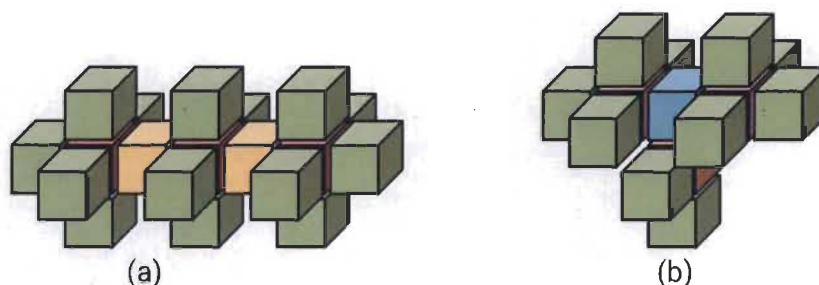


FIGURE 5.5 – L'hypothèse initiale de fonction feuille et un contre-exemple ayant plus de feuilles

La fonction conjecturée $l_{\mathfrak{G}}$ n'est donc pas la fonction $L_{\mathfrak{G}}$. Avec une recherche approfondie, on observe que plus la taille augmente, plus l'écart entre la fonction conjecturée $l_{\mathfrak{G}}$ et la fonction feuille $L_{\mathfrak{G}}$ devient grande.

Le polycube de taille 15 causant le premier contre-exemple fait lui aussi partie d'une famille construite récursivement. Plutôt que d'ajouter 6 cellules à la fois, la famille est construite en ajoutant une structure composée de 41 cellules et 28 feuilles successivement. Cet ensemble de cellules, appelé la *super-antenne* est le polycube arbre enraciné figé dont l'arborescence des degrés est $4(4(4(4(4(3(5,5))),4(4(3(5,5))))))$ qui est illustré dans la figure 5.6.

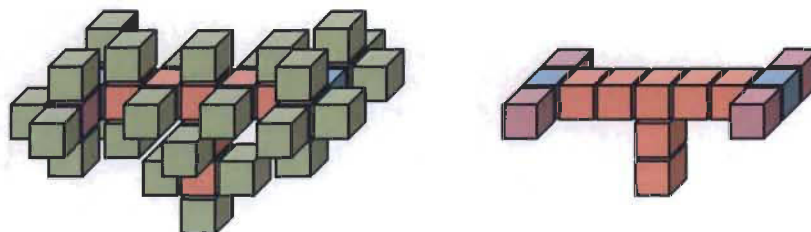


FIGURE 5.6 – La super-antenne de taille 41 ayant 28 feuilles et ses cellules intérieures

La fonction proposée ℓ_{\boxplus} , comme les fonctions ℓ_{\square} et ℓ_{\triangle} , a des cas de base et une partie récursive. La grille cubique étant plus complexe que la grille carrée, il faut une fonction accessoire f_{\boxplus} pour décrire les cas de base. La fonction ℓ_{\boxplus} est de période 41, beaucoup plus longue que dans les grilles planaires. C'est-à-dire que la valeur de $\ell_{\boxplus}(n)$ est définie à partir de $\ell_{\boxplus}(n - 41)$ et nécessite 80 valeurs initiales.

Définition 5.1.1 – ℓ_{\boxplus}

Pour des entiers positifs n , la fonction feuille potentielle ℓ_{\boxplus} est définie par

$$\ell_{\boxplus}(n) = \begin{cases} f_{\boxplus}(n) + 1 & \text{si } n \in \{6, 7, 13, 19, 25\}, \\ f_{\boxplus}(n) & \text{si } 2 \leq n \leq 40 \text{ et } n \notin \{6, 7, 13, 19, 25\}, \\ f_{\boxplus}(n - 41) + 28 & \text{si } 41 \leq n \leq 81, \\ \ell_{\boxplus}(n - 41) + 28 & \text{sinon.} \end{cases}$$

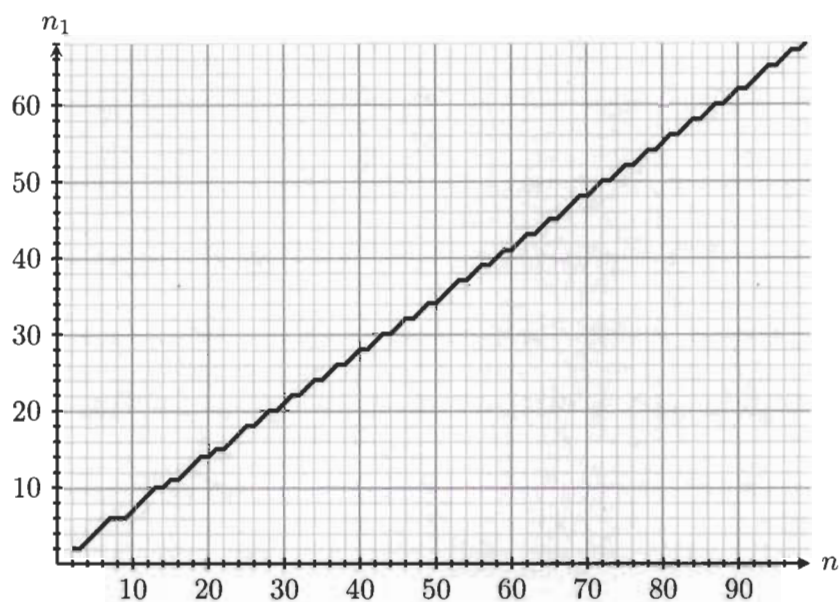
Où f_{\boxplus} est définie par

$$f_{\boxplus}(n) = \begin{cases} \left\lfloor \frac{2n+2}{3} \right\rfloor, & \text{si } 0 \leq n \leq 11, \\ \left\lfloor \frac{2n+3}{3} \right\rfloor, & \text{si } 12 \leq n \leq 27, \\ \left\lfloor \frac{2n+4}{3} \right\rfloor, & \text{si } 28 \leq n \leq 40. \end{cases}$$

Les premières valeurs de la fonction ℓ_{\boxplus} sont données dans la table 5.1 et son graphe est tracé à la figure 5.7.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$\ell_{\boxplus}(n)$	2	2	3	4	5	6	6	6	7	8	9	10	10	11	11	12	13	14

TABLE 5.1 – Valeurs de $\ell_{\boxplus}(n)$ pour n de 2 à 19

FIGURE 5.7 – Graphe de la fonction $\ell_{\sigma}(n)$

5.2 Polycubes arbre enracinés figés

De la définition 2.3.1 un polycube arbre enraciné figé est un triplet $R = (T, x, \{x, y\})$ formé d'une polycube arbre T , d'une cellule x de T qui est appelée la racine de R et d'une arête $\{x, y\}$ dont les deux cellules sont dans T appelée la direction de R . Un polycube arbre enraciné figé R est illustré en dessinant les cellules de T , en plaçant des cercles sur les faces de la racine x et des segments entre les cellules x et y .

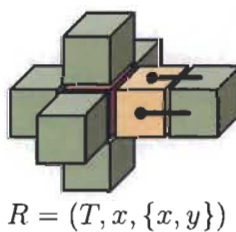


FIGURE 5.8 – Un polycube arbre enraciné figé

La greffe est une opération entre deux polycubes arbre enracinés figés dont le résultat est un polycube arbre enraciné figé. Soit $R = (T, x, \{x, y\})$ et $R' = (T', x', \{x', y'\})$ deux polycubes arbre enracinés figés. Soit τ , la translation déplaçant y' sur x . Si $z = \tau(x')$ est tel que $\{x, z\}$ est une direction libre de R , la greffe de R' sur R , notée $R \triangleleft R'$ est le polycube arbre enraciné figé

$$R \triangleleft R' = (T \cup \tau(T'), x, \{x, y\}).$$

Un exemple de greffe entre deux polycubes arbre enracinés figés est donné dans la figure 5.9.

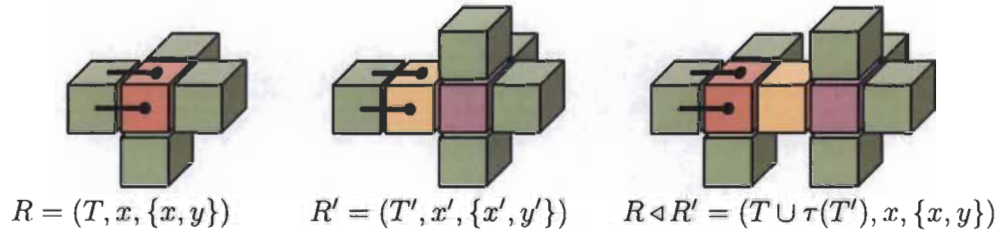


FIGURE 5.9 – Exemple de greffe de polycubes arbre enracinés figés

On définit comme à la définition 2.3.4 une branche dans un polycube arbre. Un exemple de factorisation d'un polycube arbre en une paire branche, co-branche est donné à la figure 5.10.

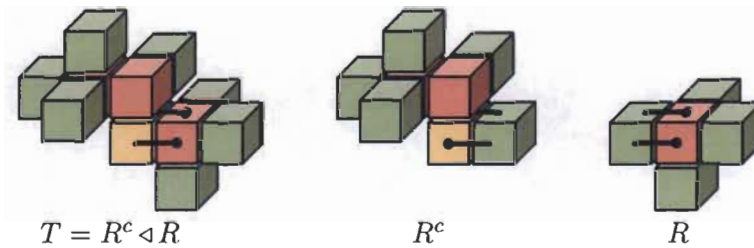


FIGURE 5.10 – Factorisation d'un polycube arbre en une paire branche, co-branche

La substitution est une opération remplaçant une branche d'un arbre T par un autre

polycube arbre enraciné figé quelconque. Si l'on a $T = R \triangleleft R^c$, alors la substitution de R par R' , noté $T' = T \triangleleft_R R'$ est bien définie par le polycube arbre $T' = R^c \triangleleft R'$ comme à la définition 2.3.5.

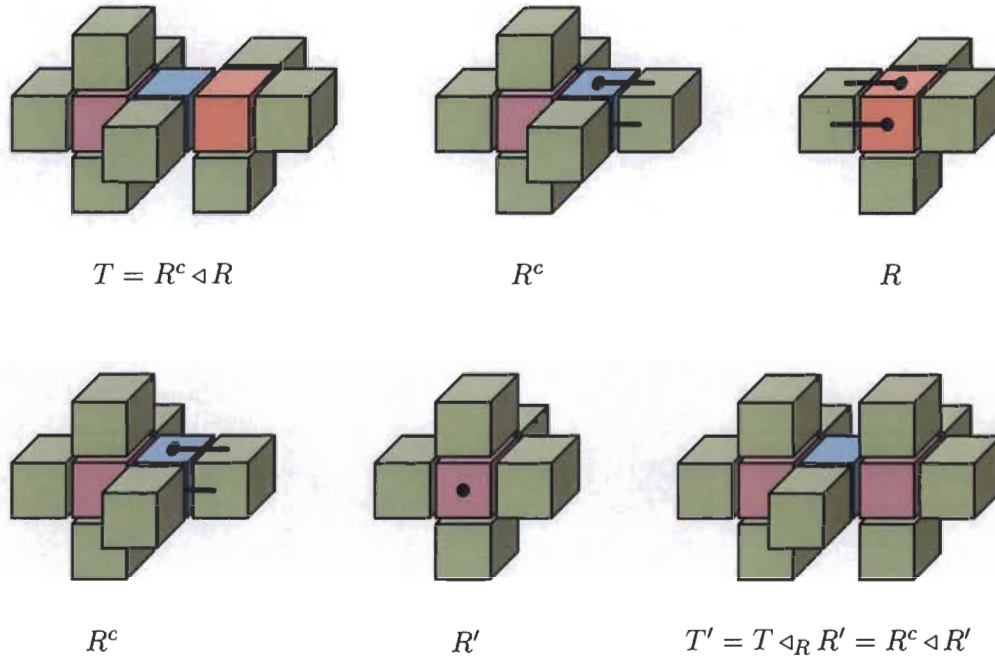


FIGURE 5.11 – Exemple de substitution de polycubes arbre enracinés figés

Si la substitution de R par R' est bien définie peu importe le polycube arbre T dont R est une branche, on dit que R est substituable par R' . De la proposition 2.3.2, un polycube arbre enraciné figé R est substituable par R' si R' est inclus dans la coque de R .

Lemme 5.2.1

Pour des entiers positifs n ,

$$\ell_{\square}(n) \leq L_{\square}(n).$$

Démonstration. On montre cette inégalité en construisant un ensemble de polycubes arbre ayant un élément de taille n pour chaque valeur de $n \geq 2$ et ayant précisément $\ell_{\square}(n)$

feuilles. Appelons $\mathcal{F}_{\mathfrak{B}}(n)$, le polycube arbre de taille n construit, on a donc

$$n_1(\mathcal{F}_{\mathfrak{B}}(n)) = \ell_{\mathfrak{B}}(n).$$

Comme la fonction $\ell_{\mathfrak{B}}(n)$ est relativement complexe, la construction de la famille $\mathcal{F}_{\mathfrak{B}}$ l'est aussi. La famille est définie par la greffe successive de polycubes arbre enracinés figés. On définit d'abord les polycubes arbre enracinés figés R_n de taille n selon leur arborescence des degrés,

$$wR_3 = 2$$

$$wR_4 = 3$$

$$wR_5 = 4$$

$$wR_{12} = 3(5, 5)$$

$$wR_{43} = 4(4(4(4(4(3(5, 5))), 4(4(3(5, 5)))))).$$

Ces polycubes sont illustrés à la figure 5.12. Pour construire l'élément $\mathcal{F}_{\mathfrak{B}}(n)$ de taille n , on pose les nombres q et r égaux au quotient et au reste de la division de n par 41, de telle sorte que $n = 41q + r$. On définit ensuite les nombres suivants

$$a = \chi(r \geq 10),$$

$$b = \chi(r \in \{1, 4, 7, 10, 11, 14, 17, 20, 23, 26, 27, 30, 33, 36, 39\}),$$

$$c = \chi(r \in \{2, 5, 8, 12, 15, 18, 21, 24, 28, 31, 34, 37, 40\}),$$

$$d = \left\lfloor \frac{(r - 10)(a + e)}{3} \right\rfloor,$$

$$e = \chi(r \geq 26).$$

On obtient l'élément voulu $\mathcal{F}_{\mathfrak{B}}(n)$ en greffant a fois R_{12} , q fois R_{43} , b fois R_3 , c fois R_4 , d fois R_5 et e fois R_{12} en ligne dans l'ordre mentionné. Des rotations sont parfois nécessaires afin de rendre les greffes bien définies. Ces greffes sont illustrées à la figure 5.12. On a donc

$$\mathcal{F}_{\mathfrak{B}}(n) = R_{12}^a \triangleleft R_{43}^q \triangleleft R_3^b \triangleleft R_4^c \triangleleft R_5^d \triangleleft R_{12}^e.$$

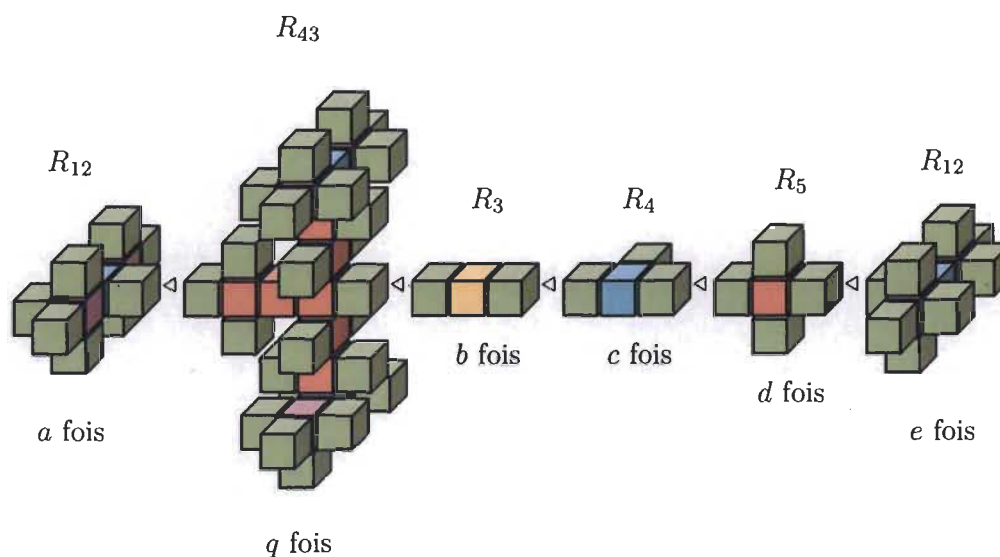


FIGURE 5.12 – Méthode de construction de la famille maximale des polycubes

Un polycube arbre T construit avec cette procédure a le nombre suivant de cellules et de feuilles

$$n(T) = 10a + 41q + 1b + 2c + 3d + 10e + 2,$$

$$n_1(T) = 7a + 28q + 0b + 1c + 2d + 7e + 2.$$

En remplaçant les définitions des nombres q, r, a, b, c, d, e , on trouve que

$$n(T) = n$$

$$n_1(T) = \ell_{\mathfrak{S}}(n(T)).$$

□

Par exemple, dans le cas où $n = 100$, on a les valeurs suivantes,

$$q = 2$$

$$r = 18$$

$$a = 1$$

$$b = 0$$

$$c = 1$$

$$d = 2$$

$$e = 0.$$

Le polycube $\mathcal{F}_{\mathfrak{G}}(100)$ est illustré dans la figure 5.13.

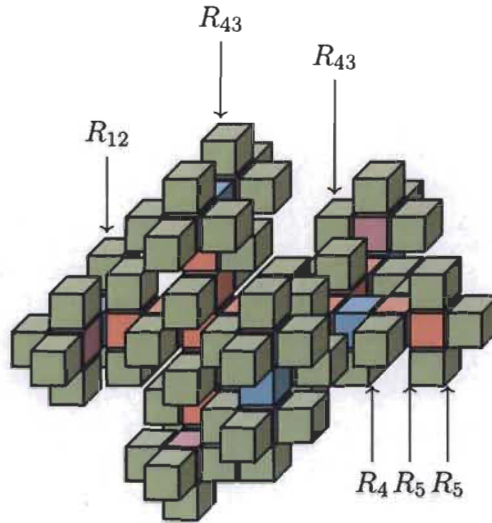


FIGURE 5.13 – Polycube pleinement feuillu de taille 100

Comme dans les sections précédentes, on utilise la fonction d'écart $\Delta_{\mathfrak{G}}$ définie en 2.2.1.

Dans le cas des polycubes, la fonction d'écart est définie par

$$\Delta(k) = \liminf_{n \geq 2} (\ell_{\mathfrak{G}}(n) - \ell_{\mathfrak{G}}(n - k)).$$

Les premières valeurs de $\Delta_{\ell_{\mathfrak{G}}}$ sont données dans la table 5.2.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
$\Delta_{\ell_{\mathfrak{G}}}(k)$	1	2	2	3	4	4	5	6	6	7	8	8	9	10	10	11	12	12

TABLE 5.2 – Premières valeurs de $\Delta_{\ell_{\mathfrak{G}}}$

Un polycubé arbre enraciné figé R est épars selon $\ell_{\mathfrak{G}}$ par un autre polycube arbre enraciné figé R' si R est substituable par R' et si

$$\Delta_{\ell_{\mathfrak{G}}}(n(R) - n(R')) \geq n_1(R) - n_1(R').$$

Autrement, R est abondant.

Il suffit donc de montrer qu'il n'existe pas de polycube arbre enraciné figé abondant à partir d'une certaine profondeur. Cette démonstration est similaire à celle pour les polyominos, les polyhexes et les polytriangles. Il suffit de faire une étude en profondeur de chaque polycube arbre enraciné figé abondant.

Contrairement aux trois grilles en dimension deux, l'étude des polycubes arbres enracinés figés abondants doit être poursuivie jusqu'à la profondeur 10 avant d'atteindre une profondeur sans branche abondante. Il n'est pas possible de compléter ce travail à la main, car il y a un trop grand nombre de cas. Il faut recourir à un programme informatique pour compléter la génération.

L'étude des premières profondeurs à la main permet d'obtenir une certaine intuition sur la structure des polycubes arbre pleinement feuillus, puisque les polycubes arbre enracinés figés abondants sont susceptibles d'apparaître dans des polycubes arbre pleinement feuillus comme branches.

Lemme 5.2.2

Pour tout entier $k \geq 11$, il n'existe aucun polycube arbre enraciné figé abondant de profondeur k .

Démonstration. On montre ce lemme en générant les polycubes arbre enracinés figés po-

tentiellement abondants et abondants pour chaque profondeur.

À la profondeur 1, les polycubes arbre enracinés figés potentiellement abondants sont les polycubes arbre enracinés figés atomiques, dont la racine est un sommet de degré 2, 3, 4, 5 ou 6. Les statistiques sur ces polycubes arbre enracinés figés sont obtenues en utilisant leur aborescence des degrés,

$$\begin{array}{ll}
 n(2) = 3 & n_1(2) = 2, \\
 n(3) = 4 & n_1(3) = 3, \\
 n(4) = 5 & n_1(4) = 4, \\
 n(5) = 6 & n_1(5) = 5, \\
 n(6) = 7 & n_1(6) = 6.
 \end{array}$$

Comme tous ces polycubes arbre enracinés figés sont substituables par \mathbf{e} , on a

$$\begin{aligned}
 \Delta(n(2) - n(\mathbf{e})) &= 0 \geq 0 = n_1(2) - n_1(\mathbf{e}), \\
 \Delta(n(3) - n(\mathbf{e})) &= 1 \geq 1 = n_1(3) - n_1(\mathbf{e}), \\
 \Delta(n(4) - n(\mathbf{e})) &= 2 \geq 2 = n_1(4) - n_1(\mathbf{e}), \\
 \Delta(n(5) - n(\mathbf{e})) &= 2 \not\geq 3 = n_1(5) - n_1(\mathbf{e}), \\
 \Delta(n(6) - n(\mathbf{e})) &= 3 \not\geq 4 = n_1(6) - n_1(\mathbf{e}).
 \end{aligned}$$

Seulement les polycubes arbre enracinés figés de degré 5 et 6 sont abondants,

$$A_1 = \{5, 6\}.$$

À la profondeur 2, il y a 4 polycubes arbre enracinés figés potentiellement abondants à isomorphismes de graphes près, leur arborescence des degrés sont

$$\{2(5), 2(6), 3(5), 3(5, 5)\}.$$

Les statistiques sur ces polycubes arbre enracinés figés sont

$$\begin{array}{ll}
 n(2(5)) = 7 & n_1(2(5)) = 5, \\
 n(2(6)) = 8 & n_1(2(6)) = 6, \\
 n(3(5)) = 8 & n_1(3(5)) = 6, \\
 n(3(5,5)) = 12 & n_1(3(5,5)) = 9.
 \end{array}$$

Comme tous ces polycubes arbre enracinés figés sont substituables par le polycube arbre enraciné figé identité, on a

$$\begin{aligned}
 \Delta(n(2(5)) - n(e)) &= 3 \geq 3 = n_1(2(5)) - n_1(e), \\
 \Delta(n(2(6)) - n(e)) &= 4 \geq 4 = n_1(2(6)) - n_1(e), \\
 \Delta(n(3(5)) - n(e)) &= 4 \geq 4 = n_1(3(5)) - n_1(e), \\
 \Delta(n(3(5,5)) - n(e)) &= 6 \not\geq 7 = n_1(3(5,5)) - n_1(e).
 \end{aligned}$$

Le seul polycube arbre enraciné figé abondant de profondeur 2 est donc celui dont l'arborescence des degrés est $3(5,5)$,

$$A_2 = \{3(5,5)\}.$$

À la profondeur 3, on a 6 polycubes arbre enracinés figés potentiellement abondants, leur aborescence des degrés est

$$\begin{aligned}
 &\{2(3(5,5)), 3(3(5,5)), 3(5,3(5,5)), \\
 &3(3(5,5), 3(5,5)), 4(3(5,5)), 4(3(5,5), 3(5,5))\}.
 \end{aligned}$$

Les statistiques sur ces polycubes arbre enracinés figés sont obtenues en étudiant les arborescences des degrés

$$\begin{array}{ll}
 n(2(3(5,5))) = 13 & n_1(2(3(5,5))) = 9, \\
 n(3(3(5,5))) = 14 & n_1(3(3(5,5))) = 10, \\
 n(3(5,3(5,5))) = 18 & n_1(3(5,3(5,5))) = 13, \\
 n(3(3(5,5), 3(5,5))) = 24 & n_1(3(3(5,5), 3(5,5))) = 17, \\
 n(4(3(5,5))) = 15 & n_1(4(3(5,5))) = 11, \\
 n(4(3(5,5), 3(5,5))) = 25 & n_1(4(3(5,5), 3(5,5))) = 18.
 \end{array}$$

En tentant de remplacer ces polycubes arbre enracinés figés par e , on obtient les inégalités suivantes

$$\begin{aligned}\Delta(n(2(3(5,5))) - n(e)) &= 7 \geq 7 = n_1(2(3(5,5))) - n(e), \\ \Delta(n(3(3(5,5))) - n(e)) &= 8 \geq 8 = n_1(3(3(5,5))) - n(e), \\ \Delta(n(3(5,3(5,5))) - n(e)) &= 10 \not\geq 11 = n_1(3(5,3(5,5))) - n(e), \\ \Delta(n(3(3(5,5),3(5,5))) - n(e)) &= 14 \not\geq 15 = n_1(3(3(5,5),3(5,5))) - n(e), \\ \Delta(n(4(3(5,5))) - n(e)) &= 8 \not\geq 9 = n_1(4(3(5,5))) - n(e), \\ \Delta(n(4(3(5,5),3(5,5))) - n(e)) &= 15 \not\geq 16 = n_1(4(3(5,5),3(5,5))) - n(e).\end{aligned}$$

Pour compléter l'élimination des polycubes arbre enracinés figés épars, on substitue les polycubes arbre enracinés figés dont l'arborescence des degrés est soit $3(5,3(5,5))$ ou $3(3(5,5),3(5,5))$ par un polycube arbre enraciné figé dont l'arborescence des degrés est 5 et on substitue le polycubes arbre enraciné figé dont l'arborescence des degrés est $4(3(5,5),3(5,5))$ par un dont l'arborescence des degrés est 6,

$$\begin{aligned}\Delta(n(3(5,3(5,5))) - n(5)) &= 8 \geq 8 = n_1(3(5,3(5,5))) - n(5), \\ \Delta(n(3(3(5,5),3(5,5))) - n(5)) &= 12 \geq 12 = n_1(3(3(5,5),3(5,5))) - n(5), \\ \Delta(n(4(3(5,5),3(5,5))) - n(6)) &= 12 \geq 12 = n_1(4(3(5,5),3(5,5))) - n(6).\end{aligned}$$

Le seul polycube arbre enraciné figé abondant de profondeur 3 est donc celui dont l'arborescence des degrés est $4(3(5,5))$,

$$A_3 = \{4(3(5,5))\}.$$

Cette procédure devient de plus en plus longue à chaque profondeur. Le travail à la main devient de plus en plus laborieux et sujet aux erreurs. Il faut donc utiliser un programme informatique pour compléter le travail. Le programme utilisé, dont l'algorithme est donné en (1), génère tous les polycubes arbre enracinés figés potentiellement abondants et abondants pour toutes les profondeurs. En résumé, l'algorithme parcourt tous les polycubes arbre enracinés figés de profondeur $p - 1$ et les greffe sur un polycube arbre enraciné figé atomique pour construire les polycubes arbre enracinés figés de profondeur p . L'algorithme retire ensuite les polycubes arbre enracinés figés qui ne sont pas abondants.

Les arborescences des degrés de polycubes arbre enracinés figés abondants sont donnés selon la profondeur plus bas. Pour simplifier la notation, lorsqu'un polycube arbre enraciné figé potentiellement abondant a au moins un parent abondant, on lui donne un étiquette de la forme a_i ou b_i . Les arborescences des degrés des polycubes arbre enracinés figés abondants sont donc,

$$A_1 = \{5, 6\},$$

$$A_2 = \{a_2 = 3(5, 5)\},$$

$$A_3 = \{a_3 = 4(a_2)\},$$

$$A_4 = \{a_4 = 4(a_3), 4(a_3, a_3), 3(5, a_3)\},$$

$$A_5 = \{a_5 = 4(a_4), b_5 = 4(a_4, a_4), c_5 = 3(5, a_4), 4(a_3, a_4)\},$$

$$A_6 = \{a_6 = 4(a_5), b_6 = 4(b_5), c_6 = 5(c_5), 4(a_3, a_5), 4(a_4, a_5)\},$$

$$A_7 = \{a_7 = 4(a_4, b_6), b_7 = 4(a_5, b_6), 4(a_6, b_6), 4(b_6, b_6), 4(a_3, b_6), 3(5, c_6), 3(b_5, c_6), 4(a_3, a_6)\},$$

$$A_8 = \{a_8 = 4(a_7), b_8 = 4(b_7), 3(5, a_7)\},$$

$$A_9 = \{a_9 = 4(a_8), 4(b_6, b_8), 4(b_6, a_8), 4(a_3, a_8), 4(a_3, b_8), 4(a_4, a_8)\},$$

$$A_{10} = \{4(a_3, a_9), 4(a_9, b_6)\},$$

$$A_{11} = \emptyset.$$

Le programme génère en tout 83946 polycubes arbre enracinés figés potentiellement abondants à translation près, desquels seulement 2670 sont abondants. Afin de vérifier si un polycube arbre enraciné figé R est substituable par un autre polycube arbre enraciné figé R' , le programme vérifie si R' est inclus dans la coque de R . Pour tester si un polycube arbre enraciné figé R est abondant, le programme le compare à toutes les autres branches abondantes, vérifie qu'elles sont substituables et que la définition d'abondance est respectée selon la fonction d'écart. \square

Pour comprendre la structure des polycubes arbre enracinés figés abondants, on utilise un graphe de descendance permettant de construire tous les polycubes arbre enracinés figés abondants. Si un polycube arbre enraciné figé R est obtenu par la greffe des polycubes arbre enracinés figés R_i sur une cellule de degré d , alors on trace une arête orientée de

Algorithme 1 Génération des branches abondantes

```

1: fonction BRANCHESABONDANTES( $p$  : profondeur) : Branches abondantes de profon-
   deur  $p$   $A_p$ 
2:    $A[1] \leftarrow \{\text{polycubes enracinés figés atomiques de degré 5 et 6}\}$ 
3:   pour  $i \leftarrow 2, 3, \dots, p$  faire
4:      $A[i] \leftarrow \emptyset$ 
5:     pour  $j \leftarrow 1, 2, \dots, d$  faire
6:       pour chaque polycube arbre enraciné figé atomique de degré  $j$ ,  $R_j$  faire
7:         pour chaque polycube arbre enraciné figé abondant  $R$  dans  $A[i-1]$ 
           faire
8:            $R_j \leftarrow R_j \triangleleft B$ 
9:           pour chaque sous-ensemble de  $i-1$  polycubes arbre enracinés figés
           abondant  $\{B_k\}$  dans  $\cup_{k=0}^{j-1} A[k]$  faire
10:             $R' \leftarrow R_j \triangleleft B_1 \triangleleft B_2 \triangleleft \dots \triangleleft B_{j-1}$ 
11:            si  $R'$  est abondante alors
12:               $A[i] \leftarrow A[i] \cup R'$ 
13:            fin si
14:          fin pour
15:        fin pour
16:      fin pour
17:    fin pour
18:  retourner  $A$ 
19: fin fonction

```

chaque R_i vers R et on étiquette le sommet R avec un d . Ce graphe est dessiné à la figure 5.14 lorsqu'un polycube arbre enraciné figé contient deux fois le même enfant, on dessine deux arêtes.

Lemme 5.2.3

Pour tout entier positif n ,

$$\ell_{\mathfrak{O}}(n) \geq L_{\mathfrak{O}}(n).$$

Démonstration. Supposons le contraire, c'est-à-dire qu'il existe des nombres entiers n tels que $\ell_{\mathfrak{O}}(n) < L_{\mathfrak{O}}(n)$. Choisissons la valeur de n minimale pour laquelle cette inégalité est respectée. Il existe donc un polycube arbre T de taille n minimale ayant plus de feuilles

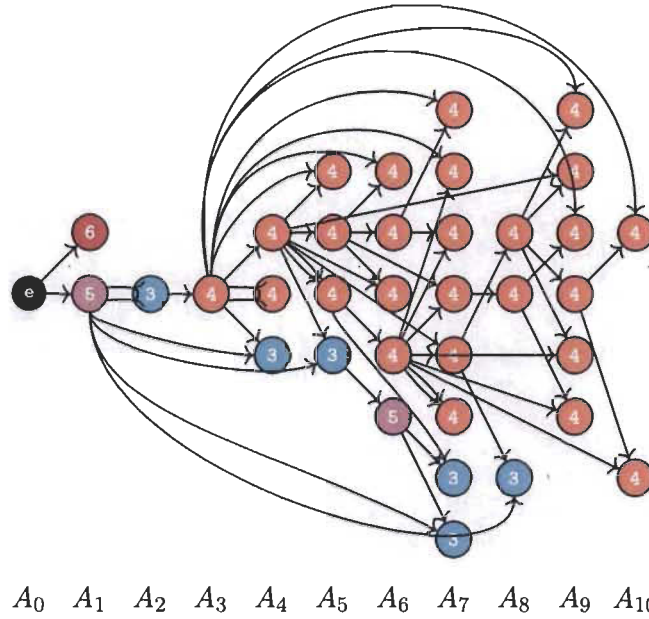


FIGURE 5.14 – Arborescence des polycubes arbre enracinés figés abondants dans les polycubes selon la profondeur

que $\ell_{\mathfrak{G}}(n)$,

$$n_1(T) > \ell_{\mathfrak{G}}(n(T)).$$

Si T contient un polycube arbre enraciné figé R comme branche qui est épars, ou de profondeur supérieure à 10, alors T contient une branche épars. Supposons que T contiennent le polycube arbre enraciné figé R épars par R' . Le polycube arbre $T' = T \triangleleft_R R'$ est tel que $n(T') < n(T)$ et $n_1(T') > \ell_{\mathfrak{G}}(n(T'))$, contredisant la minimalité de T .

Si toutes les branches de T sont de profondeur au plus 10 et sont toutes abondantes, alors on construit ces objets en greffant des polycubes arbre enracinés figés abondants sur un polycube arbre enraciné figé atomique. Un programme vérifie qu'aucun de ces objets n'a plus de feuilles que $\ell_{\mathfrak{G}}$, en utilisant un algorithme similaire à l'algorithme 1 en greffant toutes les combinaisons de polycubes arbre enracinés figés abondants sur un polycube arbre enraciné figé atomique. \square

Proposition 5.2.1

Pour tout entier positif $n \geq 2$, on a

$$\ell_{\mathfrak{G}}(n) = L_{\mathfrak{G}}(n).$$

La croissance asymptotique de $L_{\mathfrak{G}}$ est donc

$$L_{\mathfrak{G}}(n) \sim \frac{28}{41}n.$$

Démonstration. Directe des lemmes 5.2.1 et 5.2.3. □

La pente de la fonction $L_{\mathfrak{G}}$ est intéressante. Les nombres 28 et 41 ne partagent aucun facteur et ne sont que rarement rencontrés dans des suites connues. Ce ne sont ni des nombres carrés, ni des nombres cubiques, ni des nombres polygonaux. Ces deux nombres sont consécutifs dans la suite *somme des n premiers nombres premiers* A007504 sur l'OEIS,

$$0, 2, 5, 10, 17, 28, 41, 58, 77, 100, 129, 160,$$

mais il n'y a a priori aucun lien avec les polycubes.

5.3 Polycubes saturés

Comme dans les grilles précédentes, on utilise la proposition 2.4.1 qui nous assure qu'il existe deux fonctions bornant supérieurement et inférieurement la fonction $L_{\mathfrak{G}}$, qui sont

$$\begin{aligned}\overline{L}_{\mathfrak{G}}(n) &= \frac{28n + 36}{41}, \\ \underline{L}_{\mathfrak{G}}(n) &= \frac{28n - 6}{41}.\end{aligned}$$

Un polycube T est saturé s'il a $\overline{L}_{\mathfrak{G}}(n(T))$ feuilles. Une étude de la fonction $\overline{L}_{\mathfrak{G}}$ nous indique que ces polycubes arbre sont d'aire $41k + 28$ et ont $28k + 20$ feuilles pour chaque entier k . On définit la fonction d'écart saturé $\sigma_{L_{\mathfrak{G}}}$ dans le cas des polycubes pour $m > 0$ par

$$\sigma_{L_{\mathfrak{G}}}(m) = L_{\mathfrak{G}}(28 + 41k) - L_{\mathfrak{G}}(28 + 41k - m)$$

m	1	2	3	4	5	6	7	8	9	10
$\sigma_{L_{\mathfrak{D}}}(m)$	1	2	3	3	4	5	5	6	7	7

TABLE 5.3 – Valeurs de la fonction d'écart saturé dans les polycubes

pour un certain entier k tel que $28 + 41k - m \geq 2$. Les premières valeurs de cette fonction sont données dans la table 5.3.

Proposition 5.3.1

Soit T , un polycube arbre saturé. Les degrés des cellules de T satisfont

$$\deg(c) = \begin{cases} 5 & \text{si } p(c) = 1 \\ 3 & \text{si } p(c) = 2 \\ 4 & \text{si } p(c) \geq 3, \end{cases}$$

où $p(c)$ est la profondeur de c dans T .

Démonstration. On montre cette proposition en énumérant les polycubes arbre enracinés figés saturés. On utilise un algorithme similaire à la proposition 5.2.2, en remplaçant la condition d'abondance par la condition d'être potentiellement saturé. On trouve finalement que le seul polycube arbre enraciné figé qui est récursivement saturé est de profondeur 7, on l'appelle le capuchon, il est illustré à la figure 5.15 (a) et son arborescence des degrés est

$$4(4(4(4(4(3(5,5))), 4(4(3(5,5))), 4(4(3(5,5)))))).$$

Il est récursivement saturé par le polycube arbre enraciné figé dont l'arborescence des degrés est $4(4(3(5,5)))$.

C'est-à-dire que si un polycube saturé contient le capuchon, alors on peut obtenir un autre polycube saturé plus petit en greffant la branche dont l'arborescence des degrés est $4(4(3(5,5)))$ sur le capuchon. Inversement, on peut construire un polycube arbre saturé

plus grand en substituant lorsque c'est possible, une branche dont l'arborescence des degrés est $4(4(3(5,5)))$ par un capuchon.

On a donc que toutes les cellules internes à partir d'une certaine profondeur sont de degré 4 et que les cellules de profondeur 1 et 2 font toutes partie d'un capuchon, ce qui détermine leur degré selon la règle de la proposition. \square

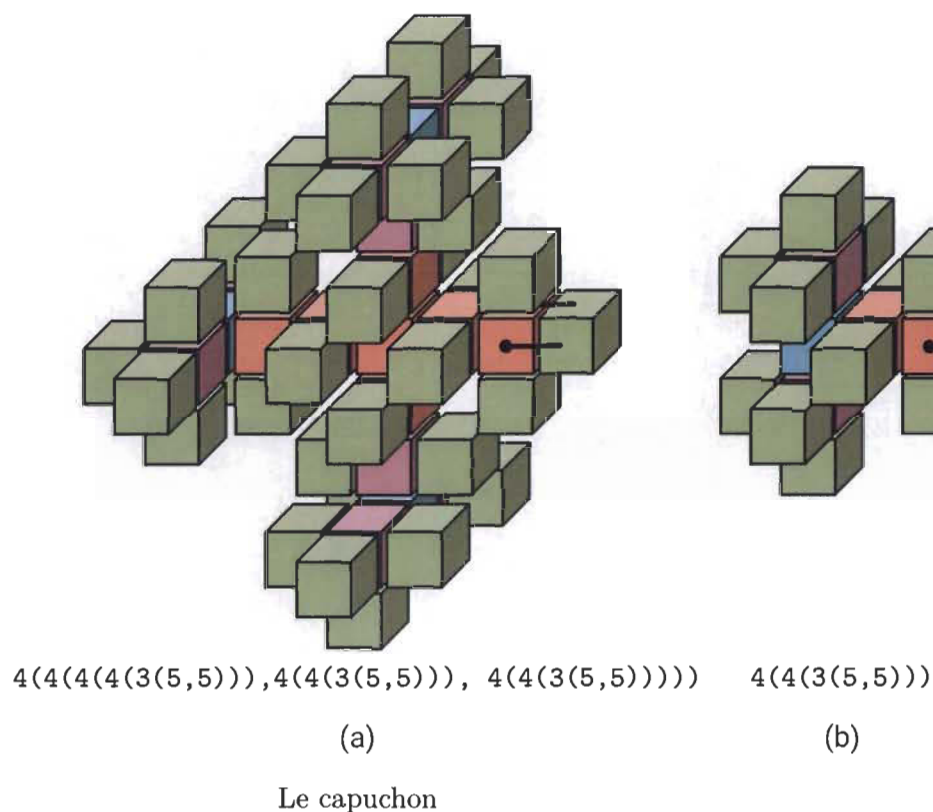


FIGURE 5.15 – L'unique polycube arbre enraciné figé récursivement saturé dans les polycubes et sa substitution

La structure des polycubes arbre saturés est moins évidente que celle des autres polyformes. Nous conjecturons qu'il existe une bijection entre les polycubes arbre saturés et des polycubes arbre dont toutes les cellules internes sont de degré 4.

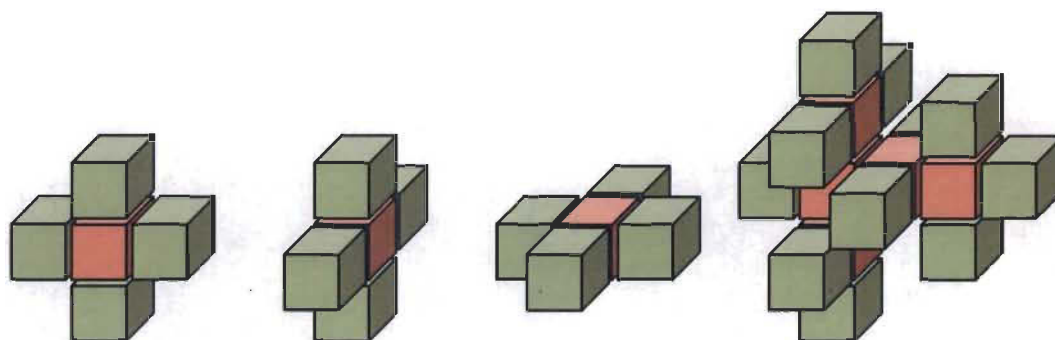


FIGURE 5.16 – Les 4-croix planaires et un 4-arbre

Définition 5.3.1

Un 4-arbre est un polycube arbre obtenu par la greffe successive de 4-croix planaires. Il existe trois 4-croix planaires à translation près. Ces 4-croix sont des polycubes de taille 5 ayant une cellule de degré 4 et 4 feuilles, toutes placées dans un même plan.

Les trois 4-croix planaires et un exemple de 4-arbre sont dessinés dans la figure 5.16.

Définition 5.3.2

La fonction ϕ est définie entre les 4-arbres ayant k cellules internes et $2k + 2$ feuilles et les polycubes saturés de taille $17(2k + 2) + 12(k) + 2$. L'image d'un 4-arbre est obtenue en remplaçant chaque cellule interne par un polycube formé de cellules de degré 4 illustré à la figure 5.17 (a) et en remplaçant les feuilles du 4-arbre par une antenne illustrée à la figure 5.17 (b).

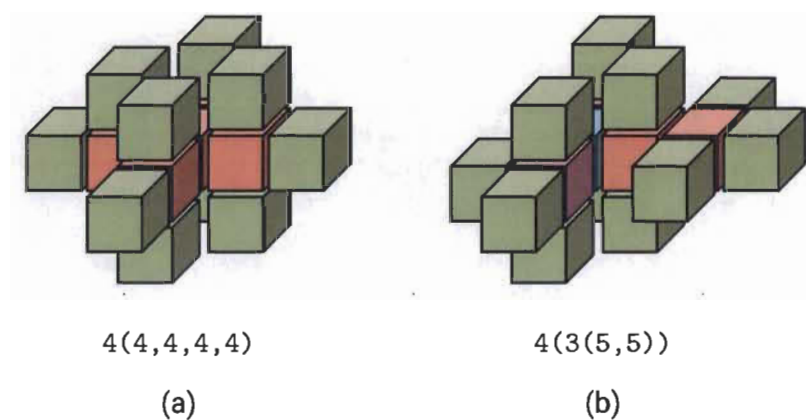
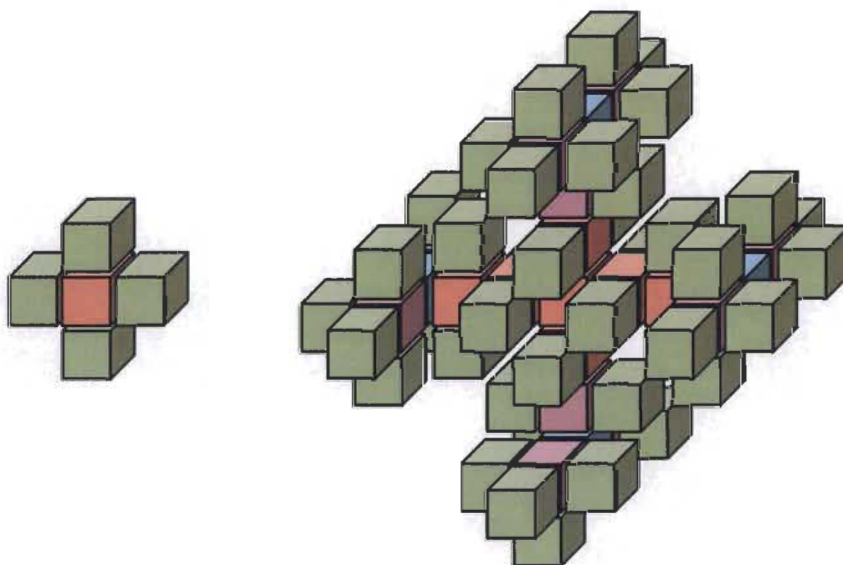
FIGURE 5.17 – Les polycubes utilisés dans la fonction ϕ 

FIGURE 5.18 – Bijection entre les 4-arbres et les polycubes saturés

Conjecture 5.3.1

La fonction ϕ entre les 4-arbres et les polycubes arbre saturés est bijective.

Il est plutôt simple de constater que la fonction ϕ est injective et que son co-domaine est bien l'ensemble des polycubes arbre saturés. L'injectivité pourrait être démontrée en

constatant que la fonction est définie par une union disjointe de cellules. Il ne peut donc pas y avoir deux 4-arbres distincts qui ont la même image. On peut aussi constater que l'image de tout 4-arbre sous la fonction ϕ est un polycube saturé en vérifiant d'abord que l'image est un arbre et qu'il a le bon nombre de feuilles et la bonne taille pour être saturé. Il n'est pas trivial de montrer la surjectivité de cette fonction, c'est-à-dire que tout polycube arbre saturé peut être obtenu en appliquant l'opérateur croix à un 4-arbre. On connaît par contre la structure des polycubes arbre enracinés figés saturés, qui est une piste de preuve pour la surjectivité.

Le nombre de 4-arbres n'est pas connu en général. Par contre, leur structure relativement simple permet de les générer à l'aide d'un programme informatique.

On définit $fl_{\sigma}(n)$ comme le nombre de polycubes pleinement feuillus saturés de taille n à translation près. Il est donc conjecturé que cette fonction soit égale au nombre de 4-arbres à translation près.

Conjecture 5.3.2

Il y a autant de 4-arbres de taille $3k + 2$ que de polycubes arbre saturés de taille $28 + 41k$.

Cette conjecture est un corollaire de la bijectivité de la fonction ϕ .

Le nombre de 4-arbres à translation près est donné dans la table 5.4. Ces valeurs ont été générées à l'aide d'un programme informatique greffant des 4-croix de façon naïve en s'assurant de maintenir une structure d'arbre.

Cette suite n'est pas une suite connue de l'OEIS.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$fl_{\sigma}(n)$	3	6	18	42	105	258	606	1440	3519	8694	21714	54858	139875	359358

TABLE 5.4 – Nombre de 4-arbres

Conclusion

On a étudié les arbres pleinement feuillus dans les polyominos, les polytriangles, les polyhexes et les polycubes. On a donc trouvé les fonctions feuilles pour toutes ces grilles. Les arbres saturés dans chacune des ces grilles ont pu être caractérisés, et parfois énumérés. Le problème des sous-arbres induit pleinement feuillus a aussi été formulé dans le cas général des graphes.

Plusieurs résultats restent à démontrer. Entre autres, on a seulement étudié les polyformes et polycubes arbre pleinement feuillus, mais si l'on étudie tous les polyformes et polycubes, est-il possible de trouver un objet pleinement feuillus qui n'est pas un arbre. Autrement dit, est-ce que pleinement feuillu entraîne automatiquement une structure d'arbre.

Une généralisation naturelle du problème nous amène à étudier des généralisations des polyformes à des dimensions supérieures. Une généralisation du carré et du cube existe dans toutes les dimensions d , appelons ces objets des d -polyhypercubes. Un problème naturel est donc de trouver les fonctions feuille des d -polyhypercubes. On peut facilement trouver des bornes sur ces fonctions feuille L_d . Un d -hypercube a $2d$ faces, donc un maximum de $2d$ voisins. En généralisant la structure formée d'une cellule de degré 2 adjacente à une cellule de degré $2d$ donnant le nombre maximal de feuilles dans les polyominos, mais n'étant pas optimale pour les polycubes, on trouve la borne suivante sur les valeurs de L_d ,

$$L_d(n) \geq \frac{d-1}{d}n.$$

Similairement, en imaginant un polyhypercube formé uniquement de cellules de degré $2d$

toutes adjacentes, on trouve la borne supérieure suivante,

$$L_d(n) \leq \frac{2d-2}{2d-1}n + 2.$$

On remarque aisément qu'un tel objet n'est jamais un arbre et que des cellules de degré $2d$ adjacentes forment nécessairement des cycles.

Le travail d'énumération des polyformes et polycubes arbre pleinement feuillus est aussi un ensemble de problèmes supplémentaires. Une génération informatique dans le cas des polyominos donne une solution partielle. L'énumération des polyhexes et polytriangles pleinement feuillus a été faite dans l'article [BdG18a].

Aucune application sérieuse n'est encore connue des polyformes et polycubes arbre pleinement feuillus. On peut construire un scénario dans lequel les polycubes arbre saturés sont utilisés. On donne deux exemples d'applications des arbres pleinement feuillus, les deux dans une forme d'architecture.

Dans le premier cas, on cherche à construire un gratte-ciel. On voudrait avoir un maximum de murs à recouvrir de fenêtres. On pourrait accomplir cet objectif en construisant le gratte-ciel en faisant une projetant verticalement un polyomino pleinement feuillus. Par exemple, la place Ville-Marie à Montréal est une extrusion de l'unique polyomino pleinement feuillu de taille 5 qui est la croix.

Autrement, imaginons une compagnie fabriquant des vaisseaux spatiaux modulaires à partir de cubes. Chaque vaisseau est donc construit en collant des cubes ensemble de façon à ce que chaque cube forme une pièce du vaisseau. Imposer une structure d'arbre au vaisseau offre plusieurs avantages. Une telle structure permet de simplifier les déplacements. En effet, dans un arbre, il existe un unique chemin entre deux sommets. Aussi, dans le cas d'une dépressurisation soudaine dans une partie du vaisseau, le nombre de portes à fermer pour sauver le reste du vaisseau est minimal.

Considérons chacune des pièces du vaisseau individuellement. Une feuille dans un polycube arbre a la propriété d'avoir 5 de ses faces sans voisins. Supposons que chacune des faces d'une feuille soit recouverte de fenêtres. Les pièces ayant 5 fenêtres sont les plus recherchées

puisqu'elles maximisent le nombre de fenêtres et donc la lumière qui y entre et le champ de vision d'un observateur dans le pièce. Comme le constructeur de vaisseaux veut maximiser ses profits, et que les pièces situés dans les feuilles offrent les plus grands avantages, il a avantage à construire un vaisseau avec un nombre maximal de feuilles. Or, un tel vaisseau est un polycube arbre saturé, puisque ce sont ces vaisseaux qui maximisent le nombre de feuilles. Dans ce scénario, peut-être voyagerons-nous un jour dans un vaisseau spatial dont la structure est un polycube arbre saturé.

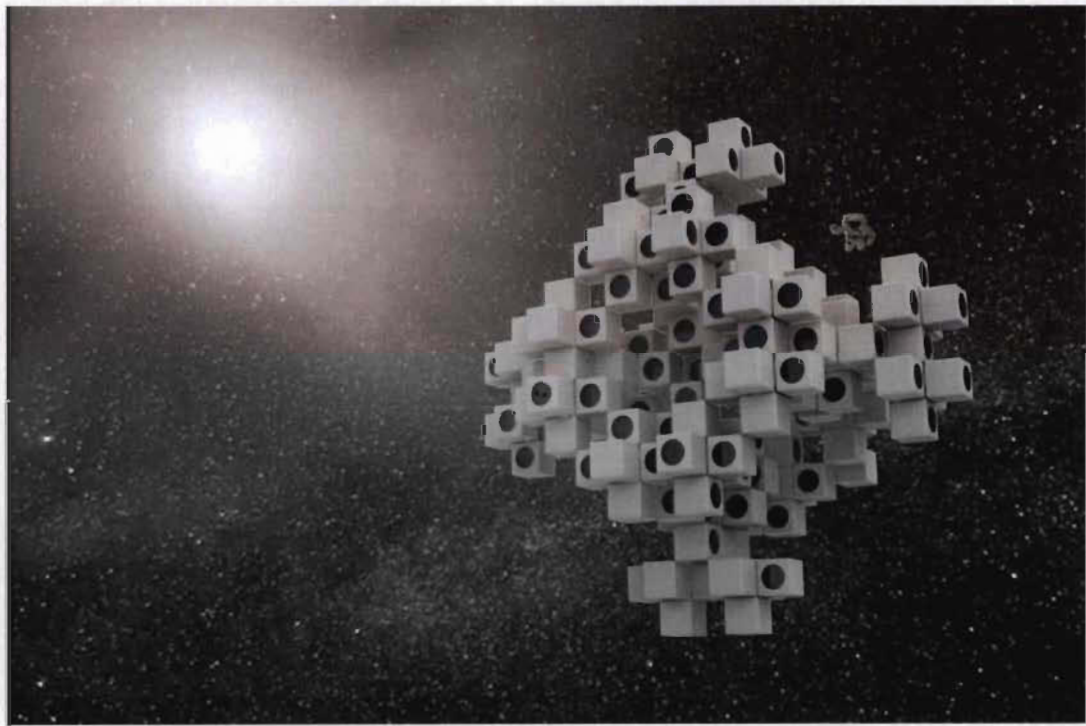


FIGURE 5.19 – Vaisseau spatial dont la structure est un polycube arbre saturé [Hé18]

Bibliographie

- [BdG17] Alexandre Blondin Massé, Julien de Carufel, and Alain Goupil. Saturated fully leafed tree-like polyforms and polycubes. Dans le livre *IWOCA*, 2017.
- [BdG18a] Alexandre Blondin Massé, Julien de Carufel, and Alain Goupil. Non saturated polyhexes and polyiamonds. Dans le livre *GASCom*, 2018.
- [BdG⁺18b] Alexandre Blondin Massé, Julien de Carufel, Alain Goupil, Mélodie Lapointe, Émile Nadeau, and Élise Vandomme. Fully leafed induced subtrees. Dans le livre *IWOCA*, 2018.
- [BdG⁺18c] Alexandre Blondin Massé, Julien de Carufel, Alain Goupil, Mélodie Lapointe, Émile Nadeau, and Élise Vandomme. Leaf realization problem, caterpillar graphs and prefix normal words. *Theoretical Computer Science*, 2018.
- [BGOR99] Mireille Bousquet-Melou, Anthony J. Guttmann, William P. Orrick, and Andrew Rechnitzer. Inversion relations, reciprocity and polyominoes. *ArXiv Mathematics e-prints*, August 1999.
- [BMR03] Mireille Bousquet-Mélou and Andrew Rechnitzer. The site-perimeter of bargraphs. *Advances in Applied Mathematics*, 31(1) :86–112, 2003.
- [BMRR06] Gill Barequet, Micha Moffe, Ares Ribó, and Günter Rote. Counting polyominoes on twisted cylinders. *Electronic journal of combinatorial number theory*, 2006.
- [de 18] Julien de Carufel. Polyformes et polycubes pleinement feuillus. Mémoire de maîtrise, Université du Québec à Trois-Rivières, 2018.

- [DGBV87] Maylis Delest, Dominic Gouyou-Beauchamps, and Bernard Vauquelin. Enumeration of parallelogram polyominoes with given bond and site perimeter. *Graphs and Combinatorics*, 3(1) :325–339, Dec 1987.
- [dWd16] Jérôme de Wouters d'Oplinter. Sur la combinatoire des polyominos serpents. Mémoire de maîtrise, Université du Québec à Trois-Rivières, 2016.
- [FGLT13] Jérôme Fortier, Alain Goupil, Jonathan Lortie, and Jérôme Tremblay. Exhaustive generation of gominoes. *Theoretical Computer Science*, 502 :76 – 87, 2013. Generation of Combinatorial Structures.
- [GCN10] Alain Goupil, Hugo Cloutier, and Fathallah Nouboud. Enumeration of polyominoes inscribed in a rectangle. 158 :2014–2023, 11 2010.
- [GCP13] Alain Goupil, Hugo Cloutier, and Marie-Eve Pellerin. Generating functions for inscribed polyominoes. *Discrete Applied Mathematics*, 161(1-2) :151–166, 2013.
- [GMM94] Giulia Galbiati, Francesco Maffioli, and Angelo Morzenti. A short note on the approximability of the maximum leaves spanning tree problem. *Information Processing Letters*, 52(1) :45–49, 1994.
- [Gol65] Solomon W. Golomb. *Polyominoes*. George Allen & Unwin Ltd, 1965.
- [Gol66] Solomon W. Golomb. Tiling with polyominoes. *Journal of Combinatorial Theory*, 1(2) :280 – 296, 1966.
- [GPdWd18] Alain Goupil, Marie-Eve Pellerin, and Jérôme de Wouters d'oplinter. Partially directed snake polyominoes. *Discrete Applied Mathematics*, 236 :223–234, 2018.
- [Hé18] Sandrine Héroux. Vaisseau spatiale, création personnelle. 1er septembre 2018.
- [Jen01] Iwan Jensen. Enumerations of lattice animals and trees. *Journal of Statistical Physics*, 102(3) :865–881, 2001.
- [Lab81] Jacques Labelle. *Théorie des graphes*. Modulo, 1981.

- [Lun72] W.F. Lunnon. Counting hexagonal and triangular polyominoes. Dans le livre RONALD C. READ, éditeur, *Graph Theory and Computing*, pages 87 – 100. Academic Press, 1972.
- [Pel15] Marie-Ève Pellerin. Énumération des polyominos d'index 0 à 2 à symétries près et des polycubes inscrits dans un prisme $2 \times 2 \times h$. Mémoire de maîtrise, Université du Québec à Trois-Rivières, 2015.
- [Red81] D. Hugh Redelmeier. Counting polyominoes : Yet another attack. *Discrete Mathematics*, 36(3) :191 – 203, 1981.
- [Rig10] Michel Rigo. *Théorie des graphes*. Université de Liège, 2009-2010.